

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО**

Факультет інформатики та обчислювальної техніки

(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління

(назва кафедри)

"На правах рукопису"

УДК 519.854.2

«До захисту допущено»

В.о. завідувача кафедри

О.А.Павлов

(підпис)

(ініціали, прізвище)

“ ” 20 19 р.

**МАГІСТЕРСЬКА ДИСЕРТАЦІЯ**

**на здобуття ступеня магістра**

за спеціальністю 126 Інформаційні системи та технології

(код та назва спеціальності)

ОПП

Інформаційні управляючі системи та технології

(код та назва спеціалізації)

на тему: Складання календарного плану виконання робіт з мінімізацією  
сумарного випередження директивного строку при виконанні робіт  
паралельними пристроями

Виконав: студент

VI курсу групи ІС-81мп

(шифр групи)

Волошин Денис Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

доц., к.т.н. Сперкач М.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

проф., д.т.н. Томашевський В.М.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

доц., к.т.н. Писаренко А.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає  
запозичень з праць інших авторів без відповідних  
посилань.

Студент

(підпис)

Київ – 2019

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти другий (магістерський) за освітньо-професійною програмою

Спеціальність 126 Інформаційні системи та технології  
(код і назва)

ОПП Інформаційні управляючі системи та технології  
(код і назва)

**ЗАТВЕРДЖУЮ**

В.о.завідувача кафедри

О.А.Павлов  
(підпис) (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 2019 р.

## ЗАВДАННЯ

**на магістерську дисертацію студенту**

Волошину Денису Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Складання календарного плану виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями

науковий керівник дисертації Сперкач Майя Олегівна, к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 28 ” жовтня 20 19 р. № 3770-с

2. Строк подання студентом дисертації “ 2 ” 12 20 19 р.

3. Об'єкт дослідження Складання календарного плану виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями

4. Перелік завдань, які потрібно розробити виконати вибір методів складання календарних планів та існуючих систем планування; розробити метод розв'язання задачі складання календарного плану та мінімізації сумарного

*випередження директивного терміну при виконанні робіт паралельними пристроями; дослідити ефективність розробленого методу побудови календарних планів; розробити програмну реалізацію запропонованого методу; здійснити експериментальне дослідження отриманих результатів.*

5. Орієнтовний перелік ілюстративного матеріалу *Схема структурна розгортання, схема структурна пакетів, схема структурна компонентів*

6. Орієнтовний перелік публікацій *тези доповідей: МОДС 2019, Матеріали III всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019); публікація: журнал «Інформатика та математичні методи в моделюванні»*

#### 7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 2 ” вересня 2019 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	<i>Прийняття проектних рішень з розробки системи складання розкладів виконання робіт на підприємстві</i>	<i>02.09</i>	
2	<i>Розробка математичних моделей та методів задачі складання розкладів з мінімізацією сумарного випередження та максимізацією моменту початку виконання робіт</i>	<i>15.09</i>	
3	<i>Результати експериментальних досліджень</i>	<i>05.10</i>	
4	<i>Розробка та опис програмного та технічного забезпечення</i>	<i>10.10</i>	
5	<i>Розробка стартап-проекту</i>	<i>27.10</i>	
7	<i>Оформлення документації</i>	<i>12.11</i>	
8	<i>Подання роботи на попередній захист</i>	<i>20.11</i>	
9	<i>Подання роботи на основний захист</i>	<i>02.12</i>	

Студент

\_\_\_\_\_  
(підпис) *Д.О. Волошин*  
(ініціали, прізвище)

Науковий керівник

\_\_\_\_\_  
(підпис) *М.О. Сперкач*  
(ініціали, прізвище)

## РЕФЕРАТ

Магістерська дисертація: 93 с., 23 рис., 24 табл., 97 джерел, 1 додаток.

**Актуальність.** Задля більш ефективного планування роботи та організації роботи необхідно використовувати різні математично-економічні методи, зокрема методи теорії розкладів. Зараз задачі теорії розкладів мають велике значення у практичних задачах. Стрімкий розвиток техніки та зв'язку все частіше викликає необхідність складання календарних планів, що пов'язані з функціонуванням сфери обслуговування промислових підприємств, освіти, транспорту тощо. Проблематика теорії розкладів включає дослідження обчислювальної складності задач, розробку наближених, евристичних або точних алгоритмів вирішення задач. Однак на практиці можливості комбінаторних алгоритмів є дуже обмеженими розмірністю задач.

Тому зараз актуальною є розробка програмного продукту для складання календарних планів виконання завдань паралельними пристроями який мінімізує сумарне випередження по директивним термінам та максимізувати момент початку виконання.

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертація була виконана на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів» (№ ДР 0117U000919).

**Мета дослідження** – збільшення ефективності функціонування виробничих систем за рахунок складання оптимальних або близьких до оптимальних за часовими критеріями календарних планів виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати аналітичний огляд існуючих моделей планування, систем планування, систем оперативно-календарного планування виробництва, методів складання календарних планів;



- розробити метод розв’язання задачі складання календарного плану та мінімізації сумарного випередження директивного терміну при виконанні робіт паралельними пристроями;
- дослідити ефективність розробленого методу побудови розкладів;
- розробити програмну реалізацію запропонованого методу;
- здійснити експериментальне дослідження отриманих результатів.

**Об’єкт дослідження** – оперативно-календарне планування підприємства.

**Предмет дослідження** – складання календарного плану виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями.

### **Наукова новизна отриманих результатів**

Розроблено алгоритм складання розкладів виконання робіт на підприємстві з використанням моделі теорії розкладів. Використання створеного алгоритму дозволяє підвищити ефективність виконання робіт за рахунок мінімізації сумарного випередження та максимізації моменту початку виконання робіт.

**Публікації.** Матеріали роботи опубліковані в тезах 14-ї Міжнародної науково-практичної конференції «МОДС 2019» та матеріалах III всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019). Стаття по темі дисертації прийнята до друку в журналі «Інформатика та математичні методи в моделюванні».

ПАРАЛЕЛЬНІ ПРИСТРОЇ, КАЛЕНДАРНИЙ ПЛАН, ДИРЕКТИВНІ ТЕРМІНИ ВИКОНАННЯ РОБІТ, МАКСИМІЗАЦІЯ МОМЕНТУ ПОЧАТКУ ВИКОНАННЯ РОБІТ, МІНІМІЗАЦІЯ СУМАРНОГО ВИПЕРЕДЖЕННЯ.

## ABSTRACT

Master's Dissertation: 93 pp., 23 fig., 24 tables, 97 sources, 1 appendixes.

**Relevance.** In order to plan and organize the work more efficiently, various mathematical and economic methods, including scheduling theory, must be used. Now the problems of scheduling theory are of great importance in practical problems. The rapid development of technology and communication is increasingly necessitated by the preparation of calendar plans related to the functioning of the industrial service sector of industrial enterprises, education, transport, etc. Problems in scheduling theory include the study of the computational complexity of problems, the development of approximate, heretical or exact algorithms for solving problems. However, in practice combinatorial algorithms are very limited in dimension.

So, it is important to develop software for scheduling tasks with parallel devices with the same performance, which will help to reduce the total penalty for violation of the directive deadline.

**Relationship with working with scientific programs, plans, topics.** The work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Igor SikorskyKyiv Polytechnic Institute» within the topic « Effective methods for solving the problems of scheduling theory» (No. DP 0117U000919).

**The purpose of the study** is to increase the efficiency of the production systems by drawing up optimal or close to it the time-based criteria of work plans with minimization of the total advance of the schedule date when performing works by parallel devices.

To achieve this goal, you must complete the following **tasks**:

- perform an analytical review of existing scheduling systems, scheduling models, scheduling methods, production and operational scheduling systems;
- develop a method for solving the task of scheduling and minimizing the total advance of the schedule date when performing works with parallel devices;
- investigate the effectiveness of the developed method of scheduling;

- develop software implementation of the proposed method;
- carry out an experimental study of the results obtained.

**The object of study** is the operational and calendar planning of enterprise.

**The subject of the study** is the job-shop scheduling with minimization of the cumulative advance of the directive term when performing work using parallel devices

### **Scientific novelty of the obtained results**

The algorithm of scheduling of work execution at the enterprise using the model of the theory of schedules was developed. Using the created algorithm allows to increase efficiency of work execution by minimizing the total advance and maximizing the moment of the beginning of the work execution.

**Publications.** The materials the work were published in the abstracts in the 14th International Scientific and Practical Conference "MODS 2019" and materials of the Third All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Technologies of Management" (ISTU-2019). The article of the master's dissertation was published in the journal "Informatics and mathematical methods in modeling".

PARALLEL DEVICES, SCHEDULE, SCHEDULE DATE, MAXIMIZING THE MOMENT OF THE BEGINNING OF THE WORK, MINIMIZING OVERALL ADVANCE.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>11</b>
<b>1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ПЛАНУВАННЯ ВИРОБНИЧИХ ПРОЦЕСІВ.....</b>	<b>13</b>
<b>1.1 Типи і методи організації виробництва.....</b>	<b>13</b>
<b>1.2 Методи управління виробництвом .....</b>	<b>16</b>
1.2.1 Автоматизовані системи управління підприємством: MRP II .....	19
1.2.2 Автоматизовані системи управління корпорацією: ERP .....	20
1.2.3 Advanced Planning & Scheduling.....	22
1.2.4 Методологія «точно в термін».....	23
<b>1.3 Класифікація задач теорії розкладів.....</b>	<b>25</b>
<b>1.4 Класифікація задач складання розкладів виконання робіт на одному пристрої .....</b>	<b>29</b>
1.4.1 Обмеження та показники ефективності.....	29
1.4.2 Властивості послідовності виконання завдань .....	31
<b>1.5 Критерії оцінки розкладів.....</b>	<b>32</b>
<b>1.6 Методи вирішення задач теорії розкладів .....</b>	<b>33</b>
<b>1.7 Змістовна постановка задачі .....</b>	<b>37</b>
<b>Висновок до розділу.....</b>	<b>38</b>
<b>2 РОЗВ'ЯЗАННЯ ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ ВИКОНАННЯ РОБІТ З МІНІМІЗАЦІЄЮ СУМАРНОГО ВИПЕРЕДЖЕННЯ ДИРЕКТИВНОГО ТЕРМІНУ ПРИ ВИКОНАННІ РОБІТ ПАРАЛЕЛЬНИМИ ПРИСТРОЯМИ.....</b>	<b>39</b>
<b>2.1 Математична постановка задачі.....</b>	<b>39</b>
<b>2.2 Дослідження властивостей задачі.....</b>	<b>39</b>
2.2.1 Характеристики робіт.....	40

2.2.2	Характеристики директивних строків .....	41
<b>2.3</b>	<b>Алгоритм розв'язання задачі .....</b>	<b>42</b>
2.3.1	Приклад розв'язання задачі .....	43
	<b>Висновок до розділу.....</b>	<b>48</b>
<b>3</b>	<b>РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ .....</b>	<b>50</b>
3.1	Кодування класів задач .....	50
3.2	Експериментальне дослідження роботи алгоритму .....	52
	<b>Висновок до розділу.....</b>	<b>54</b>
<b>4</b>	<b>ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>56</b>
4.1	Засоби розробки.....	56
4.2	Архітектура програмного забезпечення.....	61
4.3	Схема структурна пакетів.....	61
4.4	Схема структурна компонентів .....	61
4.5	Екранні форми.....	61
	<b>Висновок до розділу.....</b>	<b>63</b>
<b>5</b>	<b>РОЗРОБКА СТАРТАП-ПРОЕКТУ .....</b>	<b>64</b>
5.1	Опис ідеї проекту .....	64
5.2	Технологічний аудит ідеї проекту .....	65
5.3	Аналіз ринкових можливостей запуску стартап-проекту .....	67
5.4	Розрахунок економічної ефективності інноваційного проекту за методикою ЮНІДО .....	74
5.4.1	Визначення обсягу грошових потоків, дол. ....	75
5.4.2	Визначення норми дисконтування проекту (d).....	75

5.4.3 Визначення чистого дисконтованого доходу (ЧДД) та чистої поточної вартості (ЧПВ) .....	76
5.4.4 Визначення терміну окупності проекту .....	76
5.4.5 Індекс доходності та середньорічна рентабельність проекту .....	77
<b>Висновок до розділу.....</b>	<b>77</b>
<b>ЗАГАЛЬНІ ВИСНОВКИ .....</b>	<b>79</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>82</b>
<b>ДОДАТКИ .....</b>	<b>91</b>
<b>ДОДАТОК А .....</b>	<b>91</b>

## ВСТУП

Сучасні виробничі процеси на підприємствах мають тенденцію змінюватись із збільшенням кількості вимог, зростанням конкуренції та прихід нових гравців на виробничу арену. Саме ці та багато інших дій змушують підприємства переходити на новітні засоби автоматизації, методів керування робочим процесом, регулярно оновлювати техніку, яка використовується персоналом тощо, задля збільшення прибутку або покращення якості виробництва у порівнянні з іншими підприємствами.

Наразі задачі автоматизації робочих процесів можуть бути вирішені багатьма способами, зокрема з використанням розділів теорії розкладів.

**Актуальність.** Задля більш ефективного планування роботи та організації роботи необхідно використовувати різні математично-економічні методи, зокрема методи теорії розкладів. Зараз задачі теорії розкладів мають велике значення у практичних задачах. Стрімкий розвиток техніки та зв'язку все частіше викликає необхідність складання календарних планів, що пов'язані з функціонуванням сфери обслуговування промислових підприємств, освіти, транспорту тощо. Проблематика теорії розкладів включає дослідження обчислювальної складності задач, розробку наближених, евристичних або точних алгоритмів вирішення задач. Однак на практиці можливості комбінаторних алгоритмів є дуже обмеженими розмірністю задач.

Тому зараз актуальною є розробка програмного продукту для складання календарних планів виконання завдань паралельними пристроями який мінімізує сумарне випередження по директивним термінам та максимізувати момент початку виконання.

**Мета дослідження** – збільшення ефективності функціонування виробничих систем за рахунок складання оптимальних або близьких до оптимальних за часовими критеріями календарних планів виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати аналітичний огляд існуючих моделей планування, систем планування, систем оперативно-календарного планування виробництва, методів складання календарних планів;
- розробити метод розв’язання задачі складання календарного плану та мінімізації сумарного випередження директивного терміну при виконанні робіт паралельними пристроями;
- дослідити ефективність розробленого методу побудови розкладів;
- розробити програмну реалізацію запропонованого методу;
- здійснити експериментальне дослідження отриманих результатів.

**Об’єкт дослідження** – оперативно-календарне планування підприємства.

**Предмет дослідження** – складання календарного плану виконання робіт з мінімізацією сумарного випередження директивного строку при виконанні робіт паралельними пристроями.

### **Наукова новизна отриманих результатів**

Розроблено алгоритм складання розкладів виконання робіт на підприємстві з використанням моделі теорії розкладів. Використання створеного алгоритму дозволяє підвищити ефективність виконання робіт за рахунок мінімізації сумарного випередження та максимізації моменту початку виконання робіт.



# **1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ПЛАНУВАННЯ ВИРОБНИЧИХ ПРОЦЕСІВ**

Сучасні виробничі процеси на підприємствах мають тенденцію змінюватись із збільшенням кількості вимог, зростанням конкуренції та прихід нових гравців на виробничу арену. Саме ці та багато інших дій змушують підприємства переходити на новітні засоби автоматизації, методів керування робочим процесом, регулярно оновлювати техніку, яка використовується персоналом тощо, задля збільшення прибутку або покращення якості виробництва у порівнянні з іншими підприємствами.

Світ сьогодення переповнений великою кількістю засобів та методів керування робочим процесом та стрімко поповнюється новими. В умовах жорсткої конкуренції на ринку є необхідність швидкої реакції на нові пропозиції конкурентів.

Але більшість із них являються давно застарілими методами та засобами планування роботи підприємства, які не відповідають сучасним вимогам якості виробництва.

Наразі задачі автоматизації робочих процесів можуть бути вирішені багатьма способами, зокрема з використанням розділів теорії розкладів.

## **1.1 Типи і методи організації виробництва**

**Організація виробництва** [1] – це набір заходів, що спрямовані на поєднання знаряддя та предметами праці. Цей набір заходів спрямований на поєднання виробничих процесів на підприємстві між собою.

До форм організації виробництва також належать спеціалізація, кооперація комбінування та концентрація. Концентрація – це процес зосередження вироблення продукції на обмеженій кількості підприємств. Концентрація залежить від обсягу випуску продукції та величини потужності машин, кількості однотипного устаткування, числа та розмірів технологічно подібних виробництв. Для вимірювання концентрації використовуються значення обсягу продукції, чисельності працівників, вартості основних фондів [1].

**Комбінування** [1] – це об'єднання на підприємстві різногалузевих виробництв, що тісно пов'язаних між собою.

Існують показники, що описують рівень комбінування: кількість і вартість продукції, що можна отримати з вихідної сировини, які переробляється на комбінаті; кількість сировини та напівфабрикатів, які переробляються в подальшу продукцію на місці їх отримання; кількість побічної продукції в загальному обсязі продукції комбінату тощо [1].

**Кооперація** [1] представляє собою виробничі зв'язки підприємств, цехів, що беруть участь у виробництві. В основі лежать технологічна та подетальна форми спеціалізації. До основних показників, що визначають рівень кооперації, належать: частка деталей і напівфабрикатів; кількість підприємств, що об'єднуються з даним підприємством; кількість деталей і напівфабрикатів, які поставляються на сторону тощо.

**Спеціалізація** [1] – зосередження на підприємстві та випуску однорідної продукції або виконання окремих кроків технологічного процесу.

Необхідними умовами є стандартизація, уніфікація і типізація процесів. Стандартизація встановлює норми якості, форми деталей та готової продукції. Вона створює необхідні умови для обмеження номенклатури продукції та збільшення масштабів виробництва. Уніфікацією є зменшення існуючого різноманіття конструкцій, форм, розмірів деталей та вибір з них найбільш доцільних. Типізація цих процесів представляє собою обмеження різноманітності виробничих операцій, розроблення процесів для груп технологічно схожих деталей [1].

В умовах конкуренції у деяких випадках більш привабливою для підприємства є диверсифікація виробництва, що допускає широку різноманітність сфер діяльності завдяки розширенню номенклатури продукції. Спеціалізація на випуск обмеженої кількості продукції, що орієнтована на задоволення потреб ринку [1].

**Організація виробництва** [1] на підприємстві – форма одиничного розподілу праці.

Розрізняють три типи виробництва: одиничне, серійне, масове. Порівняльна характеристика типів виробництва надана в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика типів виробництва

Порівняльна ознака	Типи виробництва		
	одиничне	серійне	масове
Номенклатура та обсяг випуску	необмежена номенклатура, виготовлення продукції за замовленням	широка номенклатура, виготовлення продукції партіями	обмежена номенклатура, виготовлення продукції великими обсягами
Повторність випуску продукції	відсутня	періодична	постійна
Закріплення операцій за робочими місцями	відсутня	встановлюється обмежена кількість операцій	одна-дві операції на робочому місці
Розташування обладнання	за групами однорідних робочих місць	за групами для обробки конструктивно і технологічно однорідних деталей	за ходом технологічних процесів обробки
Передача предметів праці за операціями	послідовна	паралельно-послідовна	паралельна
Форма організації виробництва	технологічна	предметно-групова	прямолінійна

**Методи організації виробництва** [2] – сукупність правил ефективного поєднання основних елементів процесу виробництва в просторі і в часі.

**Метод організації індивідуального виробництва** [2]. Використовується при одиничному випуску продукції малими серіями.

Для індивідуального виробництва характерно планування ділянок по видах робіт. Послідовність розташування машин у цеху визначається маршрутом обробки деталей. Планування повинно надавати можливість переміщення деталей на малі відстані і тільки в напрямку, який веде до завершення виготовлення виробу [2].

**Метод потокового виробництва** – прогресивний метод організації виробництва, використовується в серійному і масовому виробництві, де забезпечується узгоджене виконання операцій технологічного процесу в часі та переміщення предметів праці по робочих місцях відповідно до встановленого ритмом випуску виробів. Передача предметів праці з операції на операцію здійснюється поштучно або дрібними партіями відразу ж після закінчення обробки; передбачає детальну проробку організації технічного обслуговування робочих місць [2].

## **1.2 Методи управління виробництвом**

Основою методів управління є принципи технічного та наукового розвитку підприємства та соціальні, правові і психологічні відносини між людьми [3].

Методи управління поділяються на:

- адміністративно-правові;
- економічні;
- соціально-психологічні.

**Адміністративно-правові методи управління** [3] представляють собою юридичний та адміністративний вплив на взаємовідносини людей в процесі виробництва продукції, оскільки ці відносини регулюються законами, інструкціями та положеннями.

**Економічні методи управління** [3] представляють собою методи управління, з використанням економічних законів та інтересів, а також показників.

**Соціально-психологічні методи управління** [3] дозволяють реалізувати мотиви соціальної поведінки людини, оскільки звичайні форми матеріального

заохочення поступово втрачають свою цінність. Вони передбачають визначення соціальних запитів та інтересів членів колективу, громадської думки та середовища виробництва.

**Календарне планування** [4] – це більш детальне дослідження поточного плану підприємства й занесення завдань до кожного цеху, відділу, або робітника. Планування цього випадку виконується на декаду, добу, зміну, а інколи й щогодини. Календарне планування передбачає:

- розроблення календарно-планових нормативів;
- планування руху предметів праці в просторі та часі під час виробничого процесу;
- визначення завантаження обладнання та виробничих площ;
- виконання завдань, що базуються на основі планування.

**Оперативно-календарне планування** [5] – планування діяльності компанії або підприємства, що забезпечує злагоджену роботу всіх команд, відділів, підрозділів, цехів; узгодження плану робіт та термінів; рівномірне виконання плану робіт відповідно встановленим термінам, обсягам й номенклатурі з раціональним використанням виробничих ресурсів.

Під час оперативно-календарного планування відбувається детальне планування підприємства та його підрозділів, що представляють собою виробництва, цехи, виробничі дільниці, бригади – на короткі проміжки часу. При цьому планування добре поєднується з вирішенням різноманітних питань організації їхнього виконання робіт та поточного регулювання виробництва [4].

З початку 60-х р.р., коли з'явилася можливість зберігання та аналізу великих обсягів даних (час перших операційних систем і обчислювальних комплексів для підприємств), стала розвиватися галузь розробки програмного забезпечення для підприємств [6].

Завдання планування потреб у матеріалах (Materials Requirements Planning, MRP) виявилася тією першою задачею, яка привела до створення цілої індустрії програмного забезпечення для управління підприємством.

Рішення завдання планування потреб в матеріалах реалізується за допомогою алгоритму, який також носить назву MRP-алгоритму [6].

**MRP-алгоритм** [6] – це алгоритм оптимального управління замовленнями на готову продукцію, виробництвом і запасами сировини і матеріалів.

**MRP-методологія** [6] – це реалізація MRP-алгоритму з допомогою комп'ютерної системи.

Головним завданням MRP є забезпечення гарантії наявності необхідної кількості необхідних матеріалів і комплектуючих в будь-який момент часу в рамках терміну планування, поряд з можливим зменшенням постійних запасів, а отже розвантаженням складу [6].

Однією зі складових інтегрованих інформаційних систем управління підприємством класу MRP є система планування виробничих потужностей (CRP).

Основним завданням системи CRP є перевірка здійсненності ОПП з погляду завантаження устаткування по виробничим технологічним маршрутам з урахуванням часу переналагодження, вимушених простоїв, субпідрядних робіт і т.д. [6].

Основні функції MRP систем [6]:

- опис планових одиниць і рівнів планування;
- опис специфікацій планування;
- формування основного виробничого плану графіка.

Метод MRP базується на двох найважливіших принципах:

- логіці "залежного попиту", тобто якщо є потреба в кінцевому виробі, значить є потреба у всіх його компонентах;
- забезпечення необхідними компонентами якомога пізніше, щоб рівень запасів був мінімальним.

На планування потреб у матеріалах впливає точність специфікацій і записів про стан запасів. Допущена помилка може призвести до того, що буде обчислено неправильна кількість або замовлені не ті компоненти; ця помилка не може бути виправлена до тих пір, поки не буде виявлена фізично, і часто на це йде кілька тижнів. Надійність і швидкодію ранніх систем означали, що на прогін системи йшло дуже

багато часу: від 24-х до 48 годин. Тому прогони робилися нечасто, і було неможливо перевіряти здійснимість основного плану виробництва за допомогою повторних прогонів алгоритму MRP. Тому основний план часто не виконувався і застарівав [6].

### 1.2.1 Автоматизовані системи управління підприємством: MRP II

Винахід менш дорогих обчислювальних систем реального часу і досвід роботи з MRPI привели до розробки в кінці 70-х років систем MRP в замкнутому циклі, які знайшли в даний час широке застосування [7].

Термін «замкнутий цикл» означає, що функціонування системи відбувається з урахуванням зворотного зв'язку від однієї функції до іншої. Інформація передається назад через обчислювальну систему, але при цьому ніякі дії не робляться. Ухвалення рішення про коригування плану залишається за людиною.

Планування виробничих ресурсів передбачає планування всіх ресурсів, включаючи обладнання, людські ресурси, матеріальні запаси і грошові кошти [7].

Таким чином, MRPII системи об'єднують процедури обробки замовлень на продаж, бухгалтерського обліку, закупівель та виписки рахунків-фактур з виробництвом на основі однієї бази даних реального часу.

У той же час, MRPII системи не контролюють конструкторські розробки, складання кошторису, кадри, збут і розподіл продукції, обслуговування, тобто підрозділи не об'єднані в одну систему [7].

Переваги використання систем MRPII [7]:

- поліпшити обслуговування замовників - за рахунок своєчасного виконання поставок;
- скоротити цикл виробництва і цикл виконання замовлення - отже, бізнес буде більш гнучко реагувати на попит;
- скоротити незавершене виробництво - робота не буде видаватися, поки не буде потрібно "точно до часу" для задоволення кінцевого попиту;
- значно скоротити запаси, що дозволить більш економно використовувати складські приміщення і буде потрібно менше коштів на його зберігання;

- збалансувати запаси - буде менше дефіциту і менше застарілих запасів;
- підвищити продуктивність – людські ресурси і матеріали будуть використовуватися відповідно до замовлень з меншими втратами; можна використовувати аналіз "що-якщо", щоб перевірити, чи відповідає виробництво завданням підприємства з отримання прибутку;
- створити скоординовану групу управління, яка зможе вирішувати стратегічні та оперативні питання і організувати роботу відповідно до виробленого основним планом виробництва.

Зараз у багатьох компаніях використовується система MRP II [8], що дозволяє планувати потреби підприємства в усіх виробничих ресурсах (матеріали, сировину, комплектуючі, обладнання, персонал), оперативно коригувати плани і виробничі завдання. Однак іноді виникає потреба в інформації, яку система MRP II надати не в змозі. У таких випадках системи MRP II інтегруються з системами MES, що дозволяють, поряд з виконанням інших функцій, відстежувати в реальному масштабі часу рівень завантаження обладнання, все переміщення матеріалів, деталей і вузлів, а також контролювати їх якість. Вважається, що система MES, призначена для вирішення оперативних завдань управління проектуванням, виробництвом і маркетингом, в термінах вітчизняної термінології займає проміжне положення між АСУП і АСУТП [9].

MES (manufacturing execution system, система управління виробництвом) [9] – це спеціалізована система, призначена для вирішення завдань синхронізації, координації, аналізу та оптимізації випуску продукції. Але іноді терміном MES позначають сукупність функцій автоматизованої системи, що використовуються для оперативного управління виробництвом лише на рівні цеху.

### 1.2.2 Автоматизовані системи управління корпорацією: ERP

ERP-система [10] – інформаційна система для ідентифікації та складення планів усіх ресурсів підприємства, які необхідні для продажу продукції, виробництва, закупівель деталей і їх обліку в процесі виконання замовлень клієнтів.



ERP методологія [10] представляє собою методологію планування та управління ресурсами підприємства, що є необхідними для здійснення продажів, вироблення продукції, закупівель ресурсу та обліку при виконанні клієнтських замовлень у сферах виробництва, надання послуг та дистрибуції.

Відмінності ERP від MRPII. Істотні відмінності ERP від MRP II можна виразити наступною формулою:

ERP = MRPII + реалізація всіх типів виробництва + інтегрування планування ресурсів за різними напрямками діяльності компанії + Многозвенність планування [10].

### **Характеристичні риси ERP-систем [10]**

#### **Можливість планування виробництва всіх типів в рамках однієї системи.**

Навіть на звичайному підприємстві (не кажучи вже про корпорацію) можуть співіснувати виробництва різних типів – проектного, дискретного, безперервного (процесного).

До підприємств, що працюють по безперервному процесному виробництву, можна віднести підприємства харчової, хімічної, фармацевтичної, нафтохімічної, нафтової, металургійної промисловості [10].

Підприємства, що працюють по дискретному циклу, належать до машинобудівної, легковий промисловості.

Для підтримки планування та управління всім підприємством в цілому, інформаційна система повинна "вміти" працювати з кожним з цих типів виробництв [10].

**Забезпечення багатоланкового виробничого планування.** Спочатку формуються власні плани закупівель / поставок та виробництва для кожного підприємства-ланки єдиної організаційної структури. За кожної номенклатурної одиниці, що входить під внутрішньовиробничу мережу поставок, вказується джерело (споживач) і пріоритетність поставки цієї одиниці. Потім ERP-система створює багатоланковий (агрегований) план. Перш ніж представити ці плани для затвердження, система проводить сценарну оцінку їх здійсненності. Як і в звичайних

MRPII-системах, оцінка здійсненності планів відбувається шляхом створення системою потоку замовлень залежного попиту на рівні всього виробничого об'єднання. При виявленні критичних станів плани коригуються, і лише потім надходять на затвердження [10].

**Розширення сфери інтегрованого планування ресурсів.** У класичних MRPII-системах інтегроване планування ресурсів охоплювало лише виробничі, складські, постачальницькі та збутові підрозділи підприємства. ERP-системи дозволяють залучити до сфери інтегрованого планування ресурсів всі підрозділи підприємства, так чи інакше ці ресурси використовують. Це дозволяє досягти оптимізації бізнес-операцій підприємства, а також координації дій всіх служб і підрозділів для забезпечення їх ефективної роботи [10].

**Планування та облік корпоративних фінансів.** Реалізація в ERP-системах підтримки планування ресурсів розгалуженої корпорації спричиняє необхідність посилення фінансового блоку, реалізації управління складними фінансовими потоками і можливості корпоративної консолідації. Тому в ERP-системи входять потужні системи управління корпоративними фінансами [10].

**Включення в системи потужних засобів підтримки прийняття рішень.** Сама по собі ERP-система не є інструментом для прийняття управлінських рішень, вона лише поставляє необхідну для цього інформацію. Реальну ж підтримку прийняття управлінських рішень роблять спеціальні аналітичні засоби, що вводяться в ERP-системи (зазвичай ці кошти називають OLAP – On-Line Analysis Processing) [10].

### 1.2.3 Advanced Planning & Scheduling

APS (advanced planning & scheduling – вдосконалене планування) [11] – програмне забезпечення для виробничого планування, головною особливістю якого є можливість побудови розкладу роботи обладнання в рамках всього підприємства. Отримані таким чином приватні розклади виробничих підрозділів є взаємопов'язаними з точки зору виробів та його операцій.

Основні можливості APS-систем:

- планування з точністю до секунди;
- пряме і зворотнє планування;
- багаторівнена агрегація / хвильове згладжування, що забезпечують прогнозування зверху-вниз, знизу-вгору і від середини (в обидві сторони);
- коригування прогнозу може здійснюватись як на рівні редагування числових даних, так і за допомогою миші в графічному відображенні;
- може враховуватись характер життєвого циклу продукту, при цьому беруться до уваги періоди освоєння нових продуктів і знаття з виробництва старих;
- необране число визначених користувачем одиниць вимірювання;
- безліч алгоритмів і параметрів розрахунку поповнення запасів;
- прогнозування і ведстеження матеріально-виробничих запасів;
- контроль над виконанням і повідомлення в графічному форматі, а також у вигляді звітів.

Щодо систем класу ERP, APS служить надбудовою, яка розширює і замінює їх функціональність в частині планування. При цьому APS користується інформацією, що міститься в транзакційній частині ERP (історія продажів, інформація про фактичні замовлення клієнтів, залишки товарів на складах та ін.). Як джерело інформації про стан запущених виробничих замовлень і потужностей можуть виступати MES-системи (Manufacturing Execution Systems – системи оперативного контролю за виробничим процесом, які мають власний модуль детальної оптимізації ODS, який використовується диспетчером для розрахунку і корекції поточних внутрішніх виробничих розкладів). По завершенню процесу планування APS-система передає відповідні результати, такі як замовлення на виробництво, закупівлю і переміщення, прогнози тощо, в ERP- систему [11].

#### 1.2.4 Методологія «точно в термін»

Ще однією методологією, що наразі використовується у багатьох підприємствах, є методологія «точно в термін» (Just-in-Time) [12].

Точно в термін (Just-in-Time, JIT) [12] – один з методів організації виробництва. Полягає в тому, що під час виробничого процесу необхідні для збирання деталі з'являються на виробничій ділянці в той момент, коли це необхідно, і в строго необхідній кількості. В результаті компанія, що послідовно впроваджує подібний принцип, усуває простої, мінімізує складські запаси, або може досягти зведення їх до нуля. Основні характеристики – зберігати лише необхідні запаси для виробництва; довести якість до стану «нуль дефектів»; зменшувати тривалість циклу завдяки зниженню часу оснащення, розміру черги або величини партії виробництва; поступово змінювати самі операції; здійснювати ці процеси з мінімальними витратами.

Метод «точно в термін» винайдений у 1954 р. в корпорації Toyota [13]. Він виник на основі економічних обмежень, які панували тоді в Японії. Так як у Японії було мало природних ресурсів і дуже високі ціни на нерухомість, японським фірмам не можна було допускати марнотратства, наприклад, місце збереження великих запасів зайвих товарів. Ефективність даного методу була підтверджена ще в 1973 році завдяки тривалого успіху корпорації Toyota (час нафтової кризи в Японії) [2].

Переваги даного методу [2]:

- скорочення витрат на утримання складських запасів (працівників, обладнання, оренда приміщень тощо);
- скорочення часу проведення замовлення (через зменшення розміру партії, часу переналагодження, часу простоїв);
- найкраще забезпечення матеріалами, деталями і напівфабрикатами через розташування постачальників ближче до виробників (крім того, більше робочих місць, розвиток регіонів);
- довгострокове планування для постачальників і кращий збут товарів;
- раціоналізація виробництва через спеціалізацію постачальників на замовлених товарах тощо.

Ймовірні проблеми застосування [2]:

- високі витрати на транспорт і обладнання транспортного шляху (високе навантаження на шляху, можливість пробок);
- висока залежність від одного постачальника (при недотриманні термінів поставок можливі виробничі втрати);
- висока залежність від дотримання якості матеріалів, що поставляються (витрати на вхідний контроль, рекламації);
- необхідний постійний інформаційний обмін (обов'язок підтвердження фінансового стану постачальника і виробника);
- високі штрафи для постачальників через а недотримання термінів поставок, висока залежність від одного споживача;
- необхідність для постачальників розміщувати виробництва і склади ближче до споживача;
- великі втрати у постачальників в кризових ситуаціях.

У даній роботі буде розглянута розробка технології складання розкладів з використанням методології «точно в термін», що належить до MES-систем, що вирішують завдання управління виробництвом та є частковим випадком ERP-систем, оскільки маємо справу з плануванням та управлінням корпоративних фінансів та реалізацією виробничих проектів. Розроблена система дозволить вирішувати задачі мінімізації сумарного випередження завдань у компанії з максимально пізнім моментом запуску, використовуючи декілька пристроїв різної конфігурації [2].

Існує ряд задач для організації управління виробництвом і необхідно використати різні методи, у тому числі методи теорії розкладів.

### **1.3 Класифікація задач теорії розкладів**

Теорія розкладів займає важливе місце в дослідженні операцій. Вона досліджує методи та алгоритми впорядкування робіт, засобів тощо.

Задачі впорядкування можуть виникати у випадках, де є вибір послідовності виконання робіт, наприклад, у транспортній сфері, на підприємстві, у школах тощо.

Термін «впорядкування» відрізняється від «складання розкладів», адже, це визначення черги виконання операцій одним пристроєм, де під складанням розкладів мається на увазі визначення послідовності виконання декількома пристроями [14].

В основі теорії розкладів лежить поняття «операція». Під цим розуміється задача, яку необхідно виконати. Всю сукупність операцій можна розбити на підмножини – роботи, для яких задана сукупність операцій. Для кожної сукупності визначено відношення послідовності виконання.

Задачі теорії розкладів можна розділити на дві групи [15]:

- завдання з перериваннями. У будь-який момент обслуговування завдання на пристрої може бути перервано (з можливістю завершення пізніше на тому чи іншому пристрої) заради обслуговування іншого завдання;
- завдання без переривань – кожне завдання на пристрої обслуговується від початку до кінця без переривань.

Існують різноманітні завдання теорії розкладів, деякі з них є NP-повними задачами, частина яких належить до класу поліноміальних задач, але для великої кількості завдань так і не вдалося визначити приналежність до якого-небудь класу складності. Існує твердження, що завдання, що допускає переривання, не буде складнішим, ніж завдання без переривань. Для великої кількості завдань воно справедливе, крім однієї, де для завдання без переривання є приналежність до класу поліноміальних задач, в той час коли для еквівалентної задачі з перериваннями не існує жодного доказу приналежності до якого-небудь класу складності [15].

За типом шуканого значення [15]:

- завдання узгодження. В цьому випадку основна увага цих завдань приділяється визначенню тривалості виконання робіт, часу надходження та іншим параметрам;
- завдання упорядкування. У цих завданнях вже задано розподілення робіт за виконавцями (машинами або працівниками), та визначені всі параметри робіт (тривалість виконання, директивний термін тощо). Необхідно скласти розклад виконання робіт кожним виконавцем;

- завдання розподілу. У цьому випадку відбувається визначення оптимального розподілу робіт за виконавцями.

За типом цільової функції [15]:

- завдання з сумарними критеріями оптимізації. Завдання, в яких необхідно мінімізувати сумарне значення моментів закінчення обслуговування робіт
- багатокритеріальні задачі оптимізації. Якщо необхідно скласти оптимальне рішення з точки зору декількох цільових функцій, то такі завдання будуть називатись багатокритеріальними. Наприклад, якщо в задачі необхідно як максимізувати значення, так і час простою пристрою, то задача називається багатокритеріальною задачею;
- завдання з мінімальними та максимальними критеріями оптимізації. Відмінність таких завдань від завдань з сумарними критеріями полягає у тому, що необхідно мінімізувати не суму значень, а лише максимальне з них.
- завдання на побудову допустимого розкладу. Даний клас задач можна звести до оптимізаційних задач, ввівши спеціальну функцію штрафу, який потрібно мінімізувати.

За напрямком можна виділити чотири основні класи завдань [15]:

- open shop, відкрита лінія – для кожного завдання задана своя підмножина пристроїв, на кожному з яких воно повинно обслуговуватися деякий час. Порядок обслуговування на цих пристроях довільний. Задаються різноманітні цільові функції;
- job shop, робочий цех – для кожного завдання задана своя впорядкована підмножина пристроїв (маршрут), на яких воно повинно обслуговуватися в заданому порядку. Задаються різноманітні цільові функції;
- flow shop, потокова лінія – все пристрої впорядковані і кожне завдання проходить всі пристрої в цьому порядку. Розклад задано перестановкою завдань. Як правило, мінімізується загальний час обслуговування завдань;
- задача з директивними термінами. Для кожного завдання задано момент надходження, час обслуговування і директивний термін закінчення

обслуговування. Порядок обслуговування на пристроях довільний. Необхідно знайти розклад, що дотримується директивним термінам. При існуванні такого розкладу можна ставити задачу мінімізації числа переривань.

Можливі середовища з пристроями [15]:

- одинарний пристрій;
- ідентичні пристрої;
- паралельні пристрої із різними швидкостями;
- незв'язані паралельні пристрої;
- конвеєрний цех;
- гнучкий конвеєрний цех;
- робочий цех;
- гнучкий робочий цех;
- відкритий цех.

У даній роботі будуть використані наступні види пристроїв.

**Ідентичні паралельні пристрої [16].** Існує множина ідентичних паралельних пристроїв. Завдання потребує окремої операції і може оброблятися на будь-якому із пристроїв, що належить даній множині.

**Паралельні пристрої із різними швидкостями [16].** Існує множина паралельних пристроїв із різними швидкостями. Це середовище називають рівномірними пристроями. Якщо всі пристрої мають однакову швидкість, то середовище ідентичне попередньому.

**Незв'язані паралельні пристрої.** Це середовище є ще одним узагальненням попереднього. Паралельно працюють різні пристрої. Пристрій може обробляти завдання із заданою швидкістю. Якщо швидкості пристроїв не залежать від завдань, то середовище ідентичне попередньому [16].

У даній роботі будуть вирішуватись задачі з наступними властивостями. До системи надходить множина завдань, кожне з яких має своє визначене значення



тривалості виконання та директивного терміну. Завдання виконуються на паралельних ідентичних, пропорційних та/або незв'язних пристроях. Переривання виконання завдань не припускається. Необхідно скласти розклад виконання завдань, що задовольняє заданим критеріям.

## 1.4 Класифікація задач складання розкладів виконання робіт на одному пристрої

### 1.4.1 Обмеження та показники ефективності

Розглядаються задачі складання оптимального розкладу виконання  $N$  завдань (робіт) на одному пристрої. Всі завдання складаються тільки з однієї операції. В кожен момент часу пристрій може виконувати тільки одне завдання [17].

Класифікація задач складання розкладів виконання завдань на одному пристрої наведена на рисунку 1.1.



Рисунок 2.1 – Класифікація розглянутих задач складання розкладів виконання робіт на одному пристрої

Відомі тривалості виконання кожного із завдань, порядок виконання завдань може бути довільним або регламентуватися часовими обмеженнями на початкові або кінцеві строки їх виконання та (або) обмеженнями часткового порядку. Представляють інтерес постановки та практичні додатки до задач, в яких допускається як можливість переривання виконання завдань, так і задач, в яких так переривання недопустиме. Крім того, можуть бути задані тривалості переналадок, тобто, втрат часу при переході пристроїв від виконання одної операції до іншої. Необхідно визначити розклад виконання певної кількості завдань задля забезпечення виконання всіх обмежень на директивні терміни початку та завершення, а також часткові порядки виконання завдань та досягти екстремального значення деякого критерія оптимальності [17].

Нехай за умовою задачі не допускається розривів під час виконання завдань. Тоді задачі складання розкладів зводяться до проблеми впорядкування, тобто, до визначення в деякому розумінні оптимальної послідовності (перестановка індексів) виконання завдань. Якщо заданий момент початку виконання розкладу, то складена послідовність виконання однозначно визначає моменту початку та завершення виконання кожного завдання, а також часовий графік роботи пристрою [17].

Задачам складання розкладів виконання робіт на одному пристрої без урахування обмежень на часткові послідовності та строки виконання завдань приділялась значна увага в монографіях та періодичній літературі (див. [18–24]). Однак, найбільший практичний інтерес представляють розклади, які задовольняють певним додатковим вимогам та враховують специфіку конкретних прикладних задач [17].

Допустимий розклад в цьому випадку повинен задовольняти всім поставленим вимогам. В якості критеріїв оптимальності складання розкладу виконання робіт на одному пристрої найбільш часто зустрічаються [17]:

- мінімізація суми штрафів перевищення директивних термінів завершення виконання завдань;
- вимоги виконання всього комплексу робіт у найкоротші строки;

- мінімізація сумарних втрат на переналадки або обладнання чи втрат часу на переїзди від одного пункту обслуговування до іншого;
- комплексні техніко-економічні показники, що відображають ефективність складеного розкладу вартісних показниках.

#### 1.4.2 Властивості послідовності виконання завдань

Під впорядкуванням розуміється розташування елементів у певній послідовності [17].

Порядком, чи частковим порядком, на множині елементів називається бінарне відношення на множині (визначене деякою множиною), що задовольняє умови:

- рефлексивності;
- транзитивності;
- антисиметричності.

Відношення строгого порядку називається досконалим строгим порядком, якщо для будь-яких неспівпадаючих двох елементів має місце одна з двох умов: або перший елемент знаходиться перед другим, або другий перед першим. Якщо не виконується жодна з пар умов, то елементи являються несумісними [17].

Множина елементів, на якій задано відношення часткового порядку, називається частково впорядкованою або лінійно впорядкованою множиною. В лінійно впорядкованій множині поняття найменшого та мінімального, а також найбільшого та максимального, співпадають. Прикладом частково або лінійно впорядкованої множини є множина дійсних чисел. Впорядкована множина має просту графічну інтерпретацію у вигляді орієнтованого графу, що не містить петель і контурів, в якому будь-яка пара двох вершин, якщо одна знаходиться перед іншою, сполучена дугою, направленою з однієї в іншу [17].

У комбінаториці перестановка – це впорядкований набір з певною кількістю чисел. Кількість чисел при цьому називається порядком перестановки. Кількість всіх можливих перестановок рівне факторіалу кількості елементів [17].

## 1.5 Критерії оцінки розкладів

Розглянемо величини, які надалі будуть використані в критеріях оцінки розкладів та співвідношення між ними.

Спочатку домовимось розрізняти вихідні та шукані величини задачі. Перші визначаються специфікою поставленої задачі, другі є результатом складання розкладу [14].

Нехай задані:

- момент готовності (момент надходження) роботи;
- плановий (директивний) термін її виконання. Ця величина являє собою момент, не пізніше якого повинна бути виконана дана робота;
- допустима тривалість обробки роботи.

Сума перших двох величин дорівнює значенню останньої, тому для кожної роботи достатньо задати дві з трьох означених величин.

Знаючи момент завершення роботи, можна визначити часове зміщення роботи як різницю моменту завершення та директивного терміну. Еквівалентне значення можна отримати в результаті різниці тривалості знаходження роботи в системі та допустимої тривалості її обробки [14].

Якщо часове зміщення роботи додатне – запізнення роботи буде еквівалентно зміщенню, в іншому випадку запізнення дорівнює нулю.

Часове зміщення також визначає значення випередження роботи та є абсолютним еквівалентним у разі від'ємного значення та нулю в іншому випадку [14].

Часове зміщення, запізнення та упередження оцінюють фактичний час закінчення роботи порівняно з її плановим терміном. Часове зміщення кожної роботи може мати будь-який знак. Якщо воно додатне, тобто робота завершується після планового терміну, то воно дає значення запізнення, якщо від'ємне, тобто робота завершується до планового терміну, то модуль зміщення дає значення упередження.

Часто роботи мають різну важливість, і тому, щоб адекватно відобразити цю ситуацію у критерії оцінки розкладу, для кожної роботи задається її вага, що відображає її відносну важливість. Так середньозважений час проходження задається

як сума добутків ваги та тривалості знаходження кожної роботи в системі, розділеного на кількість робіт [14].

В теорії розкладів найчастіше використовуються такі критерії: середні, середньозважені та максимальні значення:

- моментів закінчення робіт;
- тривалостей проходження;
- часового зміщення;
- запізнення;
- упередження;
- тривалості очікування.

Регулярним критерієм називається критерій, що є неспадною функцією моментів закінчення кожної роботи з набору, тобто значення не спадає на кожній наступній роботі в наборі [14].

## **1.6 Методи вирішення задач теорії розкладів**

«Часова» поведінка задач теорії розкладів виділяє їх в особливий клас, що суттєво відрізняється від «об'ємних» або кошторисних економічних задач. Якщо для економічних задач треба визначити, що і в якому обсязі треба виробляти, то для задач теорії розкладів треба визначити, коли і в якій послідовності треба виконувати задані роботи. Ця різниця в постановках задач визначає різницю в методах їхнього розв'язку. Для економічних задач існує математичний апарат, який дозволяє більш чи менш успішно їх розв'язувати. Для задач теорії розкладів математичний апарат розвинутий в значно меншій ступені. В теорії розкладів часто кожна задача вимагає індивідуального підходу [14].

Підходи до пошуку оптимального розкладу можна умовно розбити на 4 типи:

- математичне програмування;
- комбінаторний підхід;
- евристичний підхід;
- ймовірісно-статистичне моделювання.

**Математичне програмування** [14]. Основи теорії розкладів розвивалися в період, коли моделі математичного програмування (МП) почали застосовуватись для розв'язку економічних задач. Тоді виникли спроби побудувати моделі МП і для задач ТР. При цьому виникли такі труднощі. В задачах МП система обмежень описує ситуацію, коли деяка сукупність умов має виконуватись одночасно. В ТР ряд умов повинні виконуватись альтернативно (наприклад, або  $i$ -та робота виконується раніше  $j$ -ї, або навпаки).

**Комбінаторний підхід** [14]. Комбінаторний підхід зводиться до цілеспрямованої перестановки пар робіт в деякій вихідній послідовності, доки не буде одержано оптимальний (або близький до оптимального) розв'язок.

Спочатку нагадаємо визначення таких понять, як задачі класу  $P$ , ефективні алгоритми та  $NP$  – повні задачі.

Алгоритм називається ефективним, для якого кількість елементарних кроків зростає як поліном від розмірності вихідної задачі. Задачі, що мають ефективні (поліноміальні) алгоритми розв'язку, належать до класу  $P$ -задач. Будь-яку  $NP$  - повну задачу не можна розв'язати ніякими відомими поліноміальними алгоритмами [14].

$NP$  – повнота задачі дає підстави для побудови наближених або евристичних алгоритмів її розв'язку, застосування схем цілеспрямованого перебору варіантів (наприклад методом віток та границь), а також дослідження частинних випадків задачі (при цьому для деяких частинних випадків вдається істотно скоротити кількість варіантів перебору, для деяких – побудувати ефективні алгоритми).

Для  $NP$ -повних задач вводиться поняття поліноміального алгоритму їхнього розв'язку, який має задовольняти наступним умовам [14]:

- якщо при розв'язку задачі виконуються довільні аналітичні умови, то ця задача розв'язується точно (тобто може бути знайдений строго оптимальний розв'язок);
- в результаті роботи поліноміального алгоритму завжди відомо, чи знайдено оптимальний розв'язок;

- поліноміальний алгоритм є ефективним та статистично значимим, тобто при моделюванні індивідуальних задач істотна частка розв’язків є точними.

**Евристичні та ймовірнісні методи [14].** Незадовільний стан точних методів розв’язку задач теорії розкладів обумовив розробку наближених методів, які дозволяють одержувати задовільні розв’язки за порівняно невеликий час. Умовно наближені методи розділяються на евристичні та ймовірнісні.

Евристичні алгоритми базуються на так званому прийомі зниження вимог. При цьому використовуються деякі раціональні міркування без строгого обґрунтування і не гарантується, що знайдений розклад є оптимальним [14].

Типовим представником евристичних алгоритмів є метод локального пошуку. При цьому заздалегідь відома множина перестановок робіт використовується для покращення початкового розкладу до тих пір, доки покращення є можливим. Таким чином, знаходиться локальний оптимум в даному класі перестановок.

Іншим напрямком евристичних методів в теорії розкладів є визначення функцій пріоритетів. Для кожної  $i$ -ї роботи обчислюється значення функції  $f_i$  переваги та вибирається та робота, для якої  $f_i$  досягає мінімуму чи максимуму. Розглянемо деякі правила переваг [14]:

- правила SPT/LPT (Shortest/Longest processing time). Перевага віддається роботі (операції) з множини готових до виконання на пристрої, що звільнився, в якій час виконання на цьому пристрої є мінімальним/максимальним [25];
- правило SRT/LRT (Shortest/Longest remaining time). Перевага віддається роботі, для якої сума часів виконання залишкових робіт є найменшою/найбільшою [26].

Серед евристичних алгоритмів дуже важливу роль відіграють жадібні алгоритми. Вони полягають у тому, що на кожному етапі необхідно обрати з усіх можливих альтернатив ту, що забезпечить найменше значення критерію. Цей алгоритм можна застосувати як при визначенні початкового розкладу, так і на кроці його дальшого покращення [14].

Ще одним евристичним підходом є генетичні алгоритми, суть яких полягає у наступному. Маємо два допустимі розклади. За деякими встановленими правилами складається декілька розкладів, кожен з яких є «схожим» на ці два вихідні розклади. З одержаних нащадків визначаються найкращі за заданим критерієм і визначається, чи є вони кращими за «батьків». Якщо так, то по цих розкладах знову будуються нащадки і т.д. [14]

Перевагою евристичних методів є зручність їхньої реалізації на ЕОМ навіть при розв'язку громіздких задач.

Недоліки евристичних методів полягають в складності оцінювання одержаних розкладів до оптимального. Крім того, для кожного з наведених евристичних алгоритмів існують задачі, для яких застосування означених алгоритмів дає погані результати [14].

Ймовірнісні методи пов'язані з  $k$ -кратним моделюванням розкладів. Після цього вибирається найкращий розклад, який приймається за наближений розв'язок задачі.

Напрямок досліджень, наведений у даній роботі, за останні десятиліття був цікавим багатьом вченим. Аничкин А.С. та Семенов В.А. [27], описують проблематику систематизації теорії розкладів, що використовуються в різних предметних сферах. У публікації було представлено використання об'єктно-орієнтованих каркасів. Багато уваги присвячено задачам ресурсного планування. У роботі [28] описується математична модель організації та управління проектами. Згуровський М.З., Павлов О.А. та Халус О.А. довели [29], що задача складання допустимого розкладу оптимального водночас по максимізації моменту запуску та мінімізації сумарного випередження робіт є  $P$ -вирішуваною.

Первин Ю.А., Португал В.М., Семенов А.И. у роботі [30] описують методи вирішення задач календарного планування, що базуються на обмеженому переборі множини допустимих розв'язків. Це зумовлене використанням евристичних методів вирішення задач, що зменшують кількість операцій у методі перебору.



Серед методів перебору необхідно виділити метод гілок та меж [31]. Метод передбачає два етапи: алгоритм розбиття на гілки та вибір поточного елемента та визначення методу оцінки вершин та знаходження нижньої межі. Велика кількість експериментів продемонструвала, що на великих об'ємах даних алгоритм можна використовувати при вдалому виборі способі задання оцінок. Але у даному випадку є проблема у визначенні зв'язку оцінок та критерія якості.

Евристичні методи розв'язання задач навів Мельник О.О. у роботі [32] на прикладі багаторівневої системи планування з урахуванням налагодження приладів та обмеженими ресурсами з наведенням практичних прикладів.

Евристичні методи базуються на зниженні вимог. Методи доцільно використовувати у задачах з випадковими вхідними даними, з відсутньою математичною моделлю та у випадках відсутності точних методів вирішення задачі. Більш детально про евристичні методи наведено у працях [33–35].

Підмножиною евристичних методів є генетичні алгоритми [36–43]. Призначені для розв'язку задач оптимізації та моделювання послідовного підбору та перенесення даних на модель, що нагадує біологічну структуру.

Автори роботи [44] навели детальний опис математичного програмування, комбінаторного підходу, евристичного та ймовірнісного методів розв'язання задач теорії розкладів.

Подчасова Т.П., Португал В.М., Татаров В.А., Шкурба В.В. у своїй роботі [45] описували методи вирішення задач календарного планування, вказуючи на те, що при збільшенні розмірності задачі не поліноміально зростає час вирішення.

Всі ці методи та алгоритми пропонують досить оптимальне вирішення задачі, але вони не припускають використання декількох пристроїв, що можуть працювати паралельно.

## **1.7 Змістовна постановка задачі**

Необхідно розробити систему, яка дозволяє отримати ефективний розклад виконання робіт на підприємстві. Системі необхідно мати підтримку декількох

пристроїв, а також повинна надавати можливість оглянути результати здійснених раніше розв'язків задачі.

Система дозволяє задати:

- будь-яку кількість робіт;
- тривалість виконання кожної роботи;
- директивні терміни.

Вихідні дані:

- впорядкований розклад виконання робіт, що дозволяє мінімізувати сумарне випередження та максимізувати момент початку виконання робіт.

### **Висновок до розділу**

У даному розділі були описані типи і методи організації виробництва. Була здійснена порівняльна характеристика різних типів виробництва за різними ознаками: одиничного, серійного та масового. Були наведені методи управління виробництвом: економічні, адміністративно-правові та соціально-психологічні. Оскільки маємо справу з плануванням розкладу робіт, був здійснений опис календарного планування та, зокрема, оперативно-календарного планування, були наведені його основні завдання. Були дослідження наявні системи управління виробництвом, серед яких найбільш популярними в рамках задачі та їй подібній є MRP, MRPII, MES та ERP системи, були описані їх переваги та відмінності. Також, була описана методологія «точно в термін», оскільки в даній задачі необхідно максимізувати момент початку виконання робіт.

Була розглянута проблема планування процесу виконання робіт на підприємстві, був описаний її стан сьогодення. Були наведені технології та методи автоматизації процесів роботи підприємство, а саме календарного планування, зокрема оперативного. Були наведені наявні зараз системи управління виробництвом та здійснене коротке порівняння.

## **2 РОЗВ'ЯЗАННЯ ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ ВИКОНАННЯ РОБІТ З МІНІМІЗАЦІЄЮ СУМАРНОГО ВИПЕРЕДЖЕННЯ ДИРЕКТИВНОГО ТЕРМІНУ ПРИ ВИКОНАННІ РОБІТ ПАРАЛЕЛЬНИМИ ПРИСТРОЯМИ**

У роботі розглядається задача складання календарного плану виконання робіт із заданими директивними термінами та тривалістю виконання, які обробляють незалежні паралельні пристрої різної продуктивності.

Мета: мінімізувати сумарне випередження директивних термінів робіт та максимізувати момент початку виконання.

### **2.1 Математична постановка задачі**

Задано множину незалежних робіт  $J = \{1, 2, \dots, n\}$ , кожна з яких складається з однієї операції. Для роботи  $j \in J$  визначені тривалість виконання  $p_j$  та директивний термін виконання  $d_j$ . Роботи надходять до системи одночасно. Переривання робіт не допускається. Процес виконання робіт є неперервним: після виконання першої по порядку роботи одразу ж розпочинається виконання другої, і так до тих пір, доки не будуть виконані усі роботи [46].

Необхідно знайти допустимий розклад, у якому:

- момент початку виконання робіт ( $r$ ) є максимально пізнім;
- сумарне випередження моментів завершення робіт відносно директивних термінів приймає мінімальне значення.

### **2.2 Дослідження властивостей задачі**

Задачі були розбиті на підкласи задач.

Тип задачі залежить від:

- закону розподілу тривалостей;
- закону розподілу директивних строків.

Кожен розподіл має основні параметри:

- математичне сподівання (середнє значення);
- дисперсія.

Поки будемо вважати, що випадкові величини (ВВ) «тривалість роботи» та «директивний строк» мають рівномірний розподіл [47].

Відомо, що ВВ  $X$  рівномірно розподілена, якщо вона має таку функцію щільності ймовірностей [48]:

$$f(X) = \begin{cases} \frac{1}{b-a}, x \in [a; b], \\ 0, x \notin [a; b]. \end{cases}$$

Умовний запис рівномірно розподіленою ВВ  $X$ :  $X \sim U[a; b]$ , математичне сподівання дорівнює  $M(X) = \frac{b-a}{2}$ , дисперсія –  $D(X) = \frac{(b-a)^2}{12}$  [49].

Рівномірно розподілену ВВ  $X \sim U[a; b]$  можна задавати і іншим способом, задавши середнє значення  $c = \frac{a+b}{2}$  та значення напівінтервалу  $f = \frac{b-a}{2}$  (саме так і будемо надалі задавати рівномірно розподілені ВВ). Відмітимо, що напівінтервал співвідноситься із дисперсією так:

$$f = 3\sqrt{D(X)},$$

тому його також можна також вважати мірою розкиду ВВ [47].

### 2.2.1 Характеристики робіт

**Середнє значення тривалостей робіт** [47]. Оцінювати середню тривалість  $\bar{p}$  робіт як «мала», «середня» та «велика» будемо з огляду на те, як вона співвідноситься із значенням директивного строку (а саме, середнім значенням директивних строків  $\bar{d}$ ).

Пропонується класифікація задач по значенню середньої тривалості робіт  $\bar{p}$ , як це показано в таблиці 2.1.

Таблиця 2.1 – Класифікація по значенню середньої тривалості

Позначення підкласу	Інтервал значень
$S_p^-$	$\bar{p} \in [0.01\bar{d}; 0.05\bar{d}]$
$M_p^-$	$\bar{p} \in [0.06\bar{d}; 0.1\bar{d}]$
$L_p^-$	$\bar{p} \in [0.11\bar{d}; 0.15\bar{d}]$

**Примітка.** Крайні точки інтервалів підібрані з урахуванням результатів експериментів (результати, отримані для різних класів, суттєво відрізняються один від одного).

**Дисперсія тривалостей робіт** [47]. Наступним параметром, за яким проводиться класифікація задач, є дисперсія тривалості робіт, яка характеризує ступінь розкиду значень тривалостей робіт відносно середнього значення.

Значення напівінтервалу  $f_p$  для розподілу тривалостей робіт будемо задавати як відсоток від середнього значення тривалостей  $\bar{p}$ . З огляду на це, будемо розділяти задачі на три підкласи (таблиця 2.2).

Таблиця 2.2 – Класифікація задач за дисперсією тривалості

Позначення підкласу	Інтервал значень
$S_{f_p}$	$p \in [0.9\bar{p}; 1.1\bar{p}]$
$M_{f_p}$	$p \in [0.5\bar{p}; 1.5\bar{p}]$
$L_{f_p}$	$p \in [0.1\bar{p}; 1.9\bar{p}]$

### 2.2.2 Характеристики директивних строків

Останній параметр класифікації – ступінь розсіювання директивних строків [47]. Аналогічно ступеню розсіювання тривалостей, визначимо значення напівінтервалу  $f_d$  для розподілу директивних строків робіт як відсоток від середнього значення директивних строків  $\bar{d}$  (таблиця 2.3).

Таблиця 2.3 – Класифікація задач за дисперсією директивних термінів

Позначення підкласу	Інтервал значень
$S_{f_d}$	$d \in [0.8\bar{d}; 1.2\bar{d}]$
$M_{f_d}$	$d \in [0.6\bar{d}; 1.4\bar{d}]$
$L_{f_d}$	$d \in [0.4\bar{d}; 1.6\bar{d}]$

### 2.3 Алгоритм розв'язання задачі

Розглянемо частковий випадок для одного пристрою. Нехай маємо довільний допустимий розклад робіт  $J$ . Для кожної роботи  $j \in J$  є визначене значення директивного строку  $d_j$  та тривалості виконання  $p_j$  [50].

Однією із складових оптимального розкладу є виконання робіт у певному порядку, тому спочатку необхідно впорядкувати всі роботи з множини  $J$  за значенням директивного терміну у зростаючому порядку, тобто для  $j \in J$  справедливо  $d_j \leq d_{j+1}$ . Для зручності введемо додаткові позначення:

- $end$  – кінець виконання роботи;
- $T_j$  – запізнення роботи  $j$ ;
- $maxT$  – максимальне запізнення по всіх директивним термінам.

У таблиці 2.4 наведена впорядкована множина робіт з моментами початку та закінчення виконання [50].

Таблиця 2.4 – Впорядкована множина робіт

$r_j$	...	$end_j$	$r_{j+1}$	...	$end_{j+1}$	...	$r_n$	...	$end_n$
$j, d_j, p_j$			$j+1, d_{j+1}, p_{j+1}$			...	$n, d_n, p_n$		

В отриманому наборі робіт можна отримати роботи, що мають запізнення по директивним термінам, тому необхідно скорегувати моменти початку виконання,

тобто розпочати виконання раніше на значення максимального запізнення серед всіх робіт [50].

Максимальне запізнення буде рівне:

$$\max T = \max (end_j - d_j).$$

У таблиці 2.5 наведена множина робіт із компенсацією запізнення на значення  $\max T$  [50].

Таблиця 2.5 – Множина робіт із компенсацією запізнення

$r_j -$ $\max T$	...	$end_j -$ $\max T$	$r_{j+1} -$ $\max T$	...	$end_{j+1} -$ $\max T$	...	$r_n -$ $\max T$	...	$end_n -$ $\max T$
$j, d_j, p_j$			$j+1, d_{j+1}, p_{j+1}$			...	$n, d_n, p_n$		

В кінці необхідно розташувати роботи за принципом «роботи з найбільшим директивним терміном та найменшою тривалістю виконання – вкінець».

Серед всіх робіт обираємо ті, для яких  $T_j = 0$ . Отриманий набір робіт не буде брати участь у процесі обміну [50].

### 2.3.1 Приклад розв'язання задачі

Нехай маємо наступний набір робіт із визначеними директивними термінами та тривалістю виконання, що наведені у таблиці 2.6.

Таблиця 2.6 – Значення тривалості виконання та директивного терміну робіт

	Тривалість виконання ( $p$ )	Директивний термін ( $d$ )
1	10	43
2	2	33
3	3	34

Продовження таблиці 2.6

	Тривалість виконання ( $p$ )	Директивний термін ( $d$ )
4	2	36
5	5	50
6	4	20

Нехай маємо часову шкалу, на якій будуть розташовуватись роботи:

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Рисунок 2.1 – Часова шкала

Примітка. Значення були взяті з розрухунку максимального директивного терміну та сумі тривалості робіт.

Побудова початкового розкладу – сортування робіт за директивним терміном у порядку зростання.

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
	6-4/20				2-2/33		3-3/34		4-2/36		1-10/43										5-5/50						

Рисунок 2.2 – Відсортований набір робіт за директивним терміном

Компенсація моментів початку виконання на значення максимального запізнення по всім директивним термінам:  $28 - 20 = 8$ .

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
	6-4/20				2-2/33		3-3/34		4-2/36		1-10/43										5-5/50						

Рисунок 2.3 – Набір робіт із компенсацією запізнь

Процес обміну – розташування робіт за принципом «роботи з найбільшим директивним терміном та найменшою тривалістю виконання – вкінець».

Роботи без запізнення у процесу обміну не беруть участь (робота 1).

На першій ітерації поточний директивний термін дорівнює моменту закінчення виконання останньої роботи у наборі:  $d = 42$ .

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
	6-4/20																										

Рисунок 2.4 – Перша ітерація



Обираємо роботи, для яких  $d_j \geq d$  (робота 5). Серед отриманого набору робіт обираємо роботу з найменшою тривалістю виконання та ставимо в кінець:

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.5 – Друга ітерація

Поточний директивний термін дорівнює  $r_j - 1$ . Повторюємо процес, доки всі роботи не будуть розташовані.

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.6 – Третя ітерація

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.7 – Четверта ітерація

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.8 – П'ята ітерація

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.9 – Шоста ітерація

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
				6-4/20																							

Рисунок 2.10 – Сьома ітерація

Псевдокод алгоритму вирішення задачі наведено нижче.

До описаних позначень були додано *advance* та *resultWorks* для зручності, що означає випередження роботи та результуючий набір робіт відповідно.

### Псевдокод алгоритму:

READ  $J, n$

SORT  $J$  BY  $d$  ASCENDING

$r[n-1] := J[n-1].d - J[n-1].p + 1$

```

FOR  $j := 0$  TO  $J.length - 2$ 

     $r[j] := r[j + 1] - J[j].p$ 

     $end[j] := r[j] + J[j].p - 1$ 

 $maxT := 0$ 

FIND  $j$  IN  $J$ 

    WHERE  $end[j] - J[j].d$  IS MAX

    DO  $maxT := end[j] - J[j].d$ 

FOR  $j := 0$  TO  $J.length - 1$ 

     $r[j] := r[j] - maxT$ 

 $resultWorks := [], worksToOrder := J$ 

FOR  $j := 0$  TO  $J.length - 1$ 

    IF  $J[j].d = r[j] + J[j].p - 1$ 

         $resultWorks[j] := J[j]$ 

    ELSE

         $worksToOrder[j] := J[j]$ 

 $currentDeadline := r[n - 1] + J[n - 1].p - 1$ 

WHILE  $worksToOrder.length > 0$ 

     $workWithLowestExecutionTime := NULL$ 

    WHILE TRUE

         $worksWithLaterDeadline := []$ 

        FIND  $j$  IN  $worksToOrder$ 

```

```

WHERE  $currentDeadline \leq worksToOrder[j].d$ 

DO  $worksWithLaterDeadline.push(worksToOrder[j])$ 

FIND  $j$  IN  $worksWithLaterDeadline$ 

WHERE  $worksWithLaterDeadline[j].p$  IS MAX

DO  $workWithLowestExecutionTime := worksWithLaterDeadline[j]$ 

 $conflictedWork := NULL$ 

FIND  $j$  IN  $resultWorks$ 

WHERE  $currentDeadline$ 

      IN  $[r[j], currentDeadline - workWithLowestExecutionTime + 1]$ 

DO  $conflictedWork := resultWorks[j]$ 

IF  $conflictedWork$  IS NULL

  BREAK

 $currentDeadline := r[conflictedWork] - 1$ 

 $newStart := currentDeadline$ 
       $- workWithLowestExecutionTime.p + 1$ 

 $r[workWithLowestExecutionTime] := newStart$ 

 $end[workWithLowestExecutionTime] := currentDeadline$ 

 $advance[workWithLowestExecutionTime] := workWithLowestExecutionTime$ 
       $- currentDeadline$ 

 $tardiness[workWithLowestExecutionTime] := currentDeadline -$ 
       $workWithLowestExecutionTime.deadline$ 

 $result.push(workWithLowestExecutionTime)$ 

```

DELETE  $j$  IN *worksToOrder*

WHERE  $j = \text{workWithLowestExecutionTime}$

WRITE *resultWorks*

Розглянемо випадок з використанням декількох пристроїв [50].

Використаємо впорядковану множину робіт, наведену у таблиці 2.5.

Нижче наведено алгоритм розв'язання задачі [51].

**Крок 1.** Пошук пристрою, максимально пізній момент закінчення якого найбільш наближений та не більше значення  $d$ .

**Крок 2.** Пошук роботи з множини  $J$ , для якої справедлива нерівність  $d_j \geq d$ .

**Крок 3.** Якщо множина  $J$  не містить жодної роботи, що задовольняє нерівність – пошук роботи з найбільш наближеним значенням директивного терміну  $d_j$ . Інакше – серед отриманого набору робіт обираємо роботу з найменшою тривалістю виконання.

**Крок 4.** Призначаємо отриману роботу на **Кроці 3** на пристрій, отриманий на **Кроці 1**.

**Крок 5.** Якщо робота має запізнення – зменшуємо момент початку виконання, уникнувши запізнення.

**Крок 6.** Якщо множина  $J$  ще містить не впорядковані роботи – переходимо до **Кроку 1**. Інакше – до **Кроку 7**.

**Крок 7.** Для кожного пристрою  $i \in M$  необхідно уникнути простої. Тому зменшуємо моменти початку виконання  $j$ -ї роботи на максимальне допустиме значення. Перша робота для кожного пристрою не бере участь у процесі обміну [50].

### Висновок до розділу

У даному розділі була наведена постановка задачі та її математична інтерпретація. Були досліджені властивості задачі, були введені необхідні позначення для роботи алгоритму. Були наведені характеристики робіт для подальших експериментальних досліджень. Був наведений приклад розв'язання задачі на

конкретних визначених значеннях. Був наведений алгоритм її розв'язку. Роботи були впорядковані таким чином, що змогли досягти мінімізації сумарного випередження директивних термінів робіт та момент початку виконання є максимально можливим значенням.

### 3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

Для перевірки ефективності алгоритму задачі були розбиті на підкласи задач [47].

#### 3.1 Кодування класів задач

З урахуванням прийнятого підходу до класифікації задач, виділено усього 27 класів задач [47], кодування яких наведено в таблиці 3.1.

Таблиця 3.1 – Кодування класів задач

A/B/C	Середня тривалість	Дисперсія тривалості	Дисперсія директивних строків
1/1/1	«Мала»	«Мала»	«Мала»
1/1/2	«Мала»	«Мала»	«Середня»
1/1/3	«Мала»	«Мала»	«Велика»
1/2/1	«Мала»	«Середня»	«Мала»
1/2/2	«Мала»	«Середня»	«Середня»
1/2/3	«Мала»	«Середня»	«Велика»
1/3/1	«Мала»	«Велика»	«Мала»
1/3/2	«Мала»	«Велика»	«Середня»
1/3/3	«Мала»	«Велика»	«Велика»
2/1/1	«Середня»	«Мала»	«Мала»
2/1/2	«Середня»	«Мала»	«Середня»
2/1/3	«Середня»	«Мала»	«Велика»
2/2/1	«Середня»	«Середня»	«Мала»
2/2/2	«Середня»	«Середня»	«Середня»
2/2/3	«Середня»	«Середня»	«Велика»
2/3/1	«Середня»	«Велика»	«Мала»
2/3/2	«Середня»	«Велика»	«Середня»
2/3/3	«Середня»	«Велика»	«Велика»

Продовження таблиці 3.1

A/B/C	Середня тривалість	Дисперсія тривалості	Дисперсія директивних строків
3/1/1	«Велика»	«Мала»	«Мала»
3/1/2	«Велика»	«Мала»	«Середня»
3/1/3	«Велика»	«Мала»	«Велика»
3/2/1	«Велика»	«Середня»	«Мала»
3/2/2	«Велика»	«Середня»	«Середня»
3/2/3	«Велика»	«Середня»	«Велика»
3/3/1	«Велика»	«Велика»	«Мала»
3/3/2	«Велика»	«Велика»	«Середня»
3/3/3	«Велика»	«Велика»	«Велика»

Було проведено по 200 експериментів для різної кількості робіт в задачах (від 5 до 8) та для різних підкласів задач, результатами яких було відношення кількості задач, де розроблений алгоритм дав таке ж саме сумарне випередження тривалості робіт, як і повний перебір, до загальної кількості задач.

Для перевірки роботи алгоритму складання розкладу був розроблений алгоритм генерації індивідуальних задач, який генерує набори робіт, значення тривалості виконання та директивних термінів яких визначають відповідні значення дисперсій, псевдокод якого наведений нижче [47].

- 1 **Вхід:**
- 2  $n$  // Кількість робіт
- 3  $\bar{d}$  // Середнє значення директивних строків
- 4  $A/B/C$  // Тип задачі
- 5  $K$  // Матриця коефіцієнтів, що задають нижнє та верхнє значення середньої тривалості
- 6 //  $i = 1$  – «мала» робота,  $i = 2$  – «середня» робота,  $i = 3$  – «велика» робота
- 7 //  $j = 0$  – нижня межа,  $j = 1$  – верхня межа
- 8  $K^p$  // Матриця коефіцієнтів, що задають інтервал значень тривалостей в залежності від
- 9 // ступеня розсіювання

```

10      //  $i = 1$  – «мале» розсіювання,  $i = 2$  – «середнє»,  $i = 3$  – «велике»
11      //  $j = 1$  – нижня межа,  $j = 2$  – верхня межа
12       $K^d$  // Матриця коефіцієнтів, що задають інтервал значень директивних
    строків в залежності
13      // від ступеня розсіювання
15      //  $i = 1$  – «мале» розсіювання,  $i = 2$  – «середнє»,  $i = 3$  – «велике»
16      //  $j = 1$  – нижня межа,  $j = 2$  – верхня межа
17  Вихід:
18       $p$  // Вектор тривалостей
19       $a$  // Вектор директивних строків
20  // Визначення середньої тривалості
21   $\bar{p} = rand(K_{A1} \cdot \bar{d}, K_{A2} \cdot \bar{d})$ 
22  // Визначення границь інтервалу тривалостей
23   $a_p = K_{B1}^p \cdot \bar{p}$ 
24   $b_p = K_{B2}^p \cdot \bar{p}$ 
25  // Визначення границь інтервалу директивних строків
26   $a_d = K_{C1}^d \cdot \bar{d}$ 
27   $b_d = K_{C2}^d \cdot \bar{d}$ 
28  // Генерація робіт
29  for  $i := 1$  to  $n$  do
30       $p_i = rand(a_p, b_p)$ 
31       $d_i = rand(a_d, b_d)$ 

```

### 3.2 Експериментальне дослідження роботи алгоритму

Було проведено по 200 експериментів для різної кількості робіт в задачах (від 5 до 8) та для різних підкласів задач для декількох паралельних пристроїв, результатами яких було відношення кількості задач, де розроблений алгоритм дав таке ж саме сумарне випередження тривалості робіт, як і повний перебір, до загальної кількості задач [50].

Для перевірки роботи алгоритму складання розкладу був розроблений алгоритм генерації індивідуальних задач, який генерує набори робіт, значення тривалості виконання та директивних термінів яких визначають відповідні значення дисперсій.

До визначених класів задач були проведені експериментальні дослідження, результатами яких є ефективність роботи алгоритму складання розкладу по відношенню до методу повного перебору [50].



Для різної кількості робіт та класу задач отримали окремі значення відношення, графічне відображення значення яких наведені на рисунках 3.1–3.9, що дозволяє дослідити тенденцію зміни ефективності алгоритму складання розкладу при зміні кількості робіт в задачі, дисперсій тривалостей і директивних термінів робіт [47].

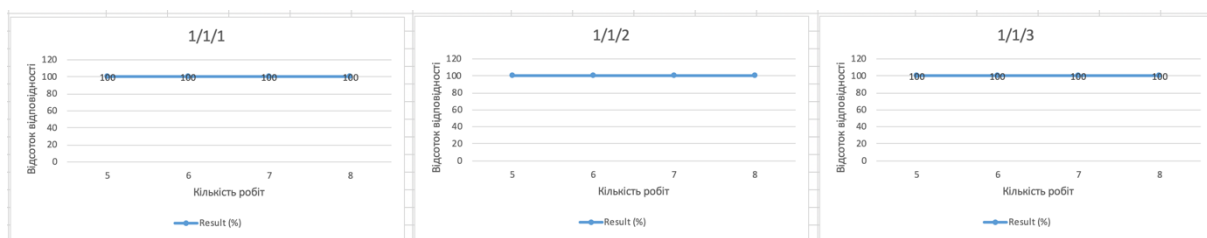


Рисунок 3.1 – Результати експериментів для задач класів 1/1/1–1/1/3

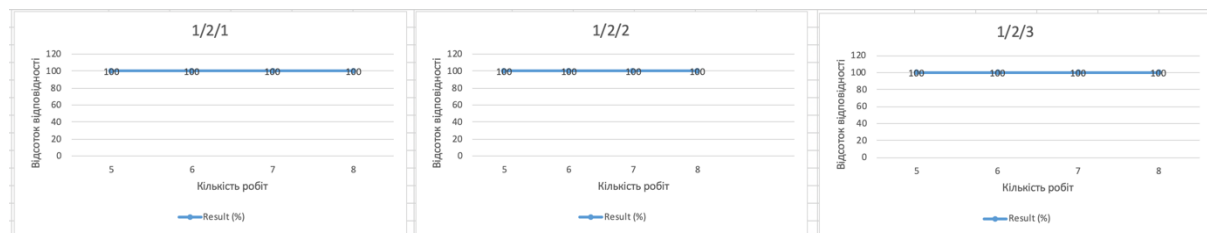


Рисунок 3.2 – Результати експериментів для задач класів 1/2/1–1/2/3

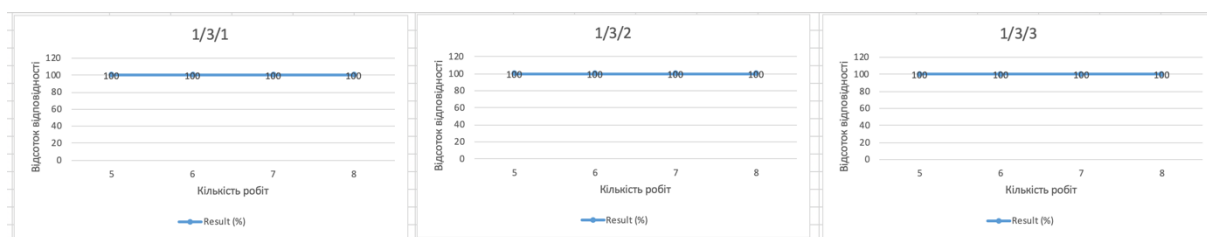


Рисунок 3.3 – Результати експериментів для задач класів 1/3/1–1/3/3

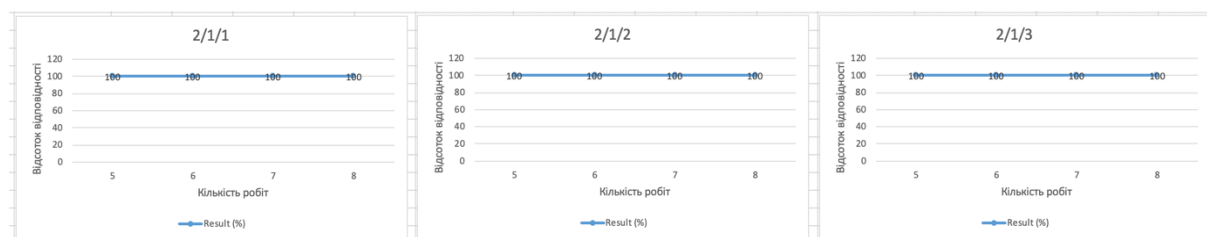


Рисунок 3.4 – Результати експериментів для задач класів 2/1/1–2/1/3

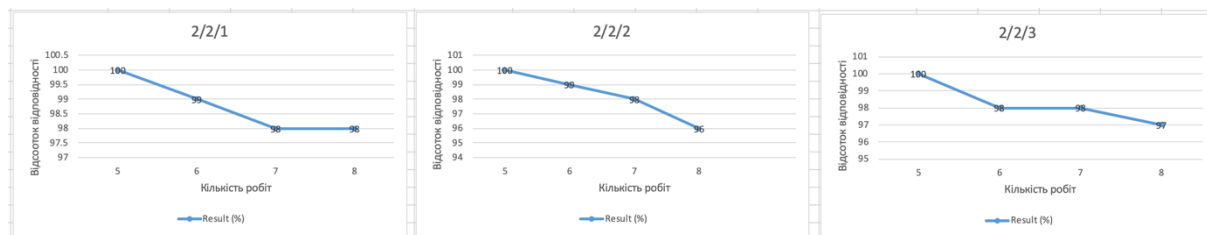


Рисунок 3.5 – Результати експериментів для задач класів 2/2/1–2/2/3

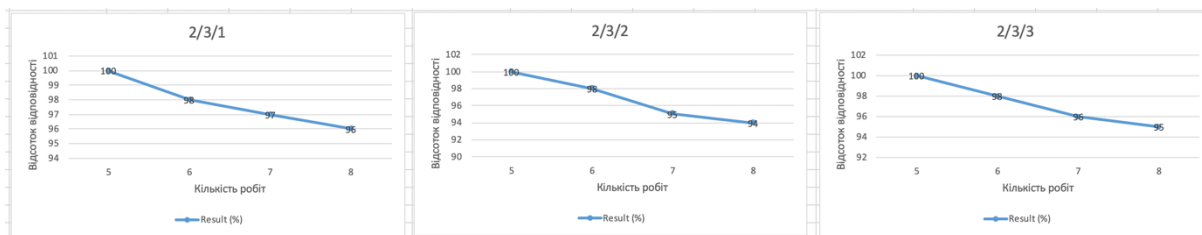


Рисунок 3.6 – Результати експериментів для задач класів 2/3/1–2/3/3

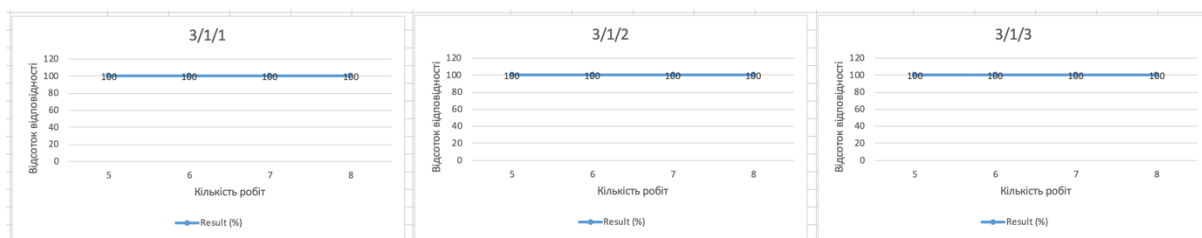


Рисунок 3.7 – Результати експериментів для задач класів 3/1/1–3/1/3

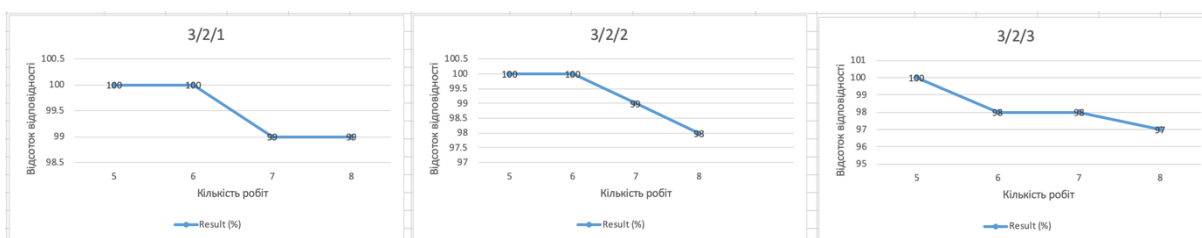


Рисунок 3.8 – Результати експериментів для задач класів 3/2/1–3/2/3

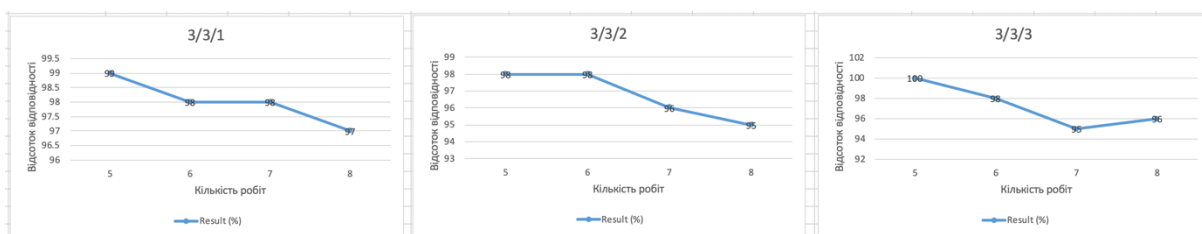


Рисунок 3.9 – Результати експериментів для задач класів 3/3/1–3/3/3

Була помічена тенденція погіршення ефективності алгоритму (у межах норми) при збільшенні значення дисперсій. Тому, алгоритм є досить ефективним в задачі, що розглядається.

### Висновок до розділу

У даному розділі було проведено експериментальне дослідження, що підтвердило ефективність розробленого алгоритму розв'язання задачі порівняно з методом повного перебору. Задля перевірки ефективності алгоритму задачі були розбиті на підкласи, кожен з яких визначає значення тривалості та директивного терміну роботи. Отримали 27 підкласів задач з різними вхідними параметрами, якими

є значення середньої тривалості виконання, дисперсії тривалості та дисперсії директивних строків робіт. Задля кращого розуміння було наведено кодування підкласів задач. Була визначена тенденція погіршення ефективності алгоритму із збільшенням значень дисперсій, але значення ефективності знаходиться у допустимих межах норми в рамках даної задачі.

## 4 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Програмний продукт було реалізовано наступними технологіями.

**Мова програмування:** JavaScript.

**JavaScript** [52] — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript [53]. Мова часто використовується для розробки веб-сторінок, що надає можливість на стороні клієнтської частини взаємодіяти з користувачем, асинхронно обмінюватися даними з сервером, керувати браузером, змінювати структуру сторінки.

Мову JavaScript можна класифікувати як прототипну мову програмування з можливістю динамічної типізації. Також JavaScript частково підтримує імперативну та частково функціональну парадигми програмування і деякі архітектурні властивості, зокрема: динамічна типізація, автоматичне керування оперативною пам'яттю, прототипне наслідування[52].

Мова JavaScript використовується для [52]:

- написання сценаріїв інтерактивних веб-сторінок;
- створення односторінкових веб-застосунків (React [54], AngularJS [55], Vue.js [56]);
- програмування на стороні сервера (Node.js [57]);
- стаціонарних застосунків (Electron [58], NW.js [59]);
- мобільних застосунків (React Native [60], Cordova [61]);
- сценаріїв в прикладному програмному забезпеченні (наприклад, в програмах зі складу Adobe Creative Suite [62] чи Apache JMeter [63]);
- всередині PDF-документів тощо.

Незважаючи на схожість імен, мова Java [64] та JavaScript мають подібні функції в стандартних бібліотеках, але є абсолютно різними мовами з різною семантикою. Синтаксис обох мов успадкований від мови C [65], але є результатом семантики та впливу мов дизайну JavaScript Self [66] та Scheme [67].

**Node.js** [57] – це платформа з відкритим вихідним кодом для запуску високопродуктивних веб-додатків, написаних на JavaScript [52]. Засновником платформи є Райан Дал (Ryan Dahl). Раніше, використовуючи JavaScript для обробки даних у браузері як третьої сторони, node.js надав можливість запустити сценарій JavaScript на сервері та надіслати результат користувачеві. Платформа Node.js змінила JavaScript на загальну мову у великій спільноті розробників.

Node.js має такі властивості [57]:

- асинхронна модель виконання запиту з однопотокових даних;
- неблокуючі введення / виведення;
- commonJS-модульна система [68];
- Google V8 JavaScript двигун [69].

**React** (раніше React.js, ReactJS) [54] – це публічна бібліотека JavaScript для створення користувацьких інтерфейсів, розроблених для вирішення проблеми часткового оновлення вмісту веб-сторінок, що виникають при розробці односторінкових застосувань. Розроблено Facebook [70], Instagram [71] та окремими спільнотами розробників.

React дозволяє розробникам створювати великі веб-додатки, які використовують дані, які змінюються з часом, не потребуючи перезавантаження сторінки. Його призначення - швидке, просте та масштабоване. React обробляє лише користувацький інтерфейс у вашій програмі. Це відповідає виду модельного виду контролера (MVC) [72] і може використовуватися з іншими бібліотеками JavaScript або з великими рамками MVC, такими як AngularJS [55]. Його також можна використовувати з доповненнями на основі React для обробки деталей без використання інтерфейсу для створення веб-додатків. Бібліотека інтерфейсу користувача React найчастіше використовується з іншими бібліотеками, такими як Redux [54].

**Одностороння передача даних.** Атрибути передаються компоненту візуалізації як атрибути тегів html. Компонент не може безпосередньо змінювати

передані атрибути, але можна змінити їх за допомогою функції зворотного виклику. Цей механізм називають «властивості донизу, події нагору» [54].

**Віртуальний DOM.** React підтримує віртуальний DOM і не покладається виключно на можливості DOM [73] браузера. Це дозволяє бібліотеці визначити, які частини DOM змінилися та визначити найкращий спосіб оновлення DOM браузера, порівняно із збереженою версією віртуального DOM. Так програміст працює зі сторінкою, вважаючи, що вся сторінка оновлена, але бібліотека самостійно визначає, які компоненти сторінки слід оновлювати [54].

**JSX.** Компоненти React зазвичай записуються у JSX [74]. Код, написаний у JSX, компілюється у виклики методу React. Розробники також можуть писати на чистому JavaScript. JSX нагадує іншу мову, створену Facebook для розширень PHP [75].

**Не лише рендеринг HTML в браузері.** React використовується не тільки для візуалізації HTML у браузерах. Наприклад, Facebook має динамічну графіку, відображену з тегами `<canvas>`, Netflix та PayPal, використовують ізоморфні навантаження для надання однорідної графіки серверу та клієнту в одному HTML [54].

**Вкладені елементи.** Кілька елементів на одному рівні повинні бути загорнуті в один елемент контейнера (наприклад, `<div>` елемент) або повернутись у вигляді масиву [54].

**Атрибути.** JSX надає декілька властивостей елементів для візуалізації з наданням HTML. Ви також можете передати спеціальні властивості компоненту. Усі властивості будуть отримані компонентом як обов'язкові [54].

**Система керування базами даних: MongoDB.**

**MongoDB** [76] – це документно-орієнтована система з відкритим кодом (СУБД) [77], яка не потребує опису схеми таблиці. MongoDB займає функціональну та зручну для користувачів нішу між швидкою та масштабованою системою ключ / значення та реляційною СКБД [78].

MongoDB підтримує зберігання документів у таких форматах, як JSON [79], має достатньо гнучку мову для запитів, може індексувати різні збережені атрибути,

ефективно забезпечує зберігання великих бінарних об'єктів та є підтримка операцій ведення журналу для зміни та додавання даних. MongoDB дозволяє працювати відповідно до парадигми Map / Reduce [80] та є підтримка конфігурації тиражування та відмовостійкості. MongoDB має вбудовані інструменти для забезпечення шардінгу (розповсюдження наборів даних на серверах на основі конкретних ключів), створення горизонтально масштабованих кластерів зберігання даних з реплікацією даних, і немає жодної точки відмови. Є автоматичне відновлення та передача навантаження невдалих вузлів. Просто додаючи нові пристрої, можна масштабувати свій кластер, не зупиняючи базу даних або перетворюючи один сервер у кластер [76].

В процесі розробки автори виходили з необхідності спеціалізації баз даних, що дозволило їм відступати від принципу єдиного розміру. Зводячи до мінімуму значення обробки транзакцій, є можливість легко розширити масштаби та підвищити ефективність. Використовувана модель зберігання документів (JSON [79] / BSON [81]) проста в кодуванні та легко керується (включаючи використання так званого «безсхемного стилю» [82]). Внутрішня кластеризація релевантних даних забезпечує додаткове підвищення продуктивності. Такий підхід є досить зручним для створення бази даних, в якій масштабування означає розгортання на декількох пристроях. Технології баз даних повинні працювати всюди – від користувацьких та віртуальних серверів до хмарних технологій [76].

### **Середовище розробки: WebStorm.**

JetBrains WebStorm [83] є інтегрованим середовищем розробки для JavaScript [52], HTML [84] та CSS [85] від компанії JetBrains [86], розробленого на платформі IntelliJ IDEA [87]. WebStorm – це спеціальна версія PhpStorm [88], яка забезпечує підмножину його можливостей. WebStorm постачається з попередньо встановленим плагіном JavaScript (наприклад, Node.js [58]), які доступні у PhpStorm безкоштовно.

WebStorm підтримує JavaScript [52], CoffeeScript [89], TypeScript [90] та Dart [91].

WebStorm забезпечує автодоповнення коду, миттєвий аналіз коду, навігацію по коду, рефакторинг [92], налагодження [93] та інтеграцію з системами управління

версіями [94]. Важливою перевагою інтегрованого середовища розробки WebStorm є те, що є можливість роботи з проектами (не тільки HTML, але і рефакторинг JavaScript-коду, вбудованого в різні файли та папки проекту). Підтримується кілька вкладень (коли сценарій JavaScript включений в HTML-документ, який містить інший HTML-код, що містить JavaScript). Ця конфігурація підтримує коректний рефакторинг [83].

### **Представлення: HTML.**

Hypertext Markup Language (HTML) [84] є стандартною мовою розмітки для створення веб-сторінок та веб-застосувань. З Cascading Style Sheets (CSS) [85] та JavaScript [52], вона утворює трійку основних технологій для World Wide Web [95].

Веб-браузер отримує документ HTML з веб-сервера або локальної пам'яті та надсилає документ у вигляді мультимедійної веб-сторінки. HTML семантично описує структуру веб-сторінки [84].

Елементи HTML – це компоненти HTML-сторінки. HTML надає інструменти для створення структурованих документів шляхом відображення структурного значення тексту, таких як заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML описуються тегами, написані за допомогою кутових дужок. Браузер не відображає теги HTML, але використовує його для інтерпретації вмісту сторінки [84].

HTML може містити програми, написані мовами сценаріїв, такими як JavaScript [52], які впливають на поведінку та вміст веб-сторінок. Увімкнення CSS визначає зовнішній вигляд та макет вмісту. World Wide Web Consortium (W3C) [96], як із стандартами HTML, так і CSS, заохочує використання CSS для явного представлення HTML з 1997 року [84].

HTML реалізує інструменти для [84].

- створення структурованих документи, відображаючи структурну організацію тексту, наприклад заголовки, абзаци, списки, таблиці та цитати;
- отримувати інформацію із всесвітньої мережі Інтернет через гіперпосилання;
- інтерактивне заповнення форми;



- додавати в текст зображення, звуки, відео та інші об'єкти.

## **4.2 Архітектура програмного забезпечення**

У ході створення продукту було використано клієнт-серверну архітектуру.

Архітектура **клієнт-сервер** [97] – це один із архітектурних шаблонів програмного забезпечення, що є домінуючою концепцією у створенні розподілених мережевих додатків і включає взаємодію та комунікацію між ними. Він містить такі основні компоненти:

- набір серверів, які надають інформацію програмам, які зверта/ться до них;
- набір клієнтів, що користуються послугами, що надаються сервером;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери незалежні один від одного. Клієнти також працюють паралельно та незалежно. Зазвичай один сервер обробляє запити від різних клієнтів одночасно. Клієнти, з іншого боку, можуть отримати доступ до того чи іншого сервера [97].

Архітектура програмного забезпечення наведена на схемі структурній розгортання, що наведена на рисунку Додатка А.1.

## **4.3 Схема структурна пакетів**

Схема структурна пакетів програмного забезпечення наведена на рисунку Додатка А.2.

## **4.4 Схема структурна компонентів**

Схема структурна компонентів зображена на рисунку Додатка А.3.

## **4.5 Екранні форми**

Система надає список раніше вирішених задач. Таблиця із списком зображена на рисунку 4.1.

Results			
#	Number of works	Sum of advances	Actions
1	6	12	<a href="#">View</a>
2	4	15	<a href="#">View</a>
3	5	15	<a href="#">View</a>
4	5	21	<a href="#">View</a>
5	6	11	<a href="#">View</a>
6	6	14	<a href="#">View</a>
7	4	9	<a href="#">View</a>
8	6	23	<a href="#">View</a>
9	4	18	<a href="#">View</a>
10	7	21	<a href="#">View</a>
11	7	17	<a href="#">View</a>

Рисунок 4.1 – Список раніше вирішених розкладів

При натисненні на кнопку «View», система відобразить детальний опис вирішення задачі, екранна форма якого відображена на рисунку 4.2.

Result of #15 schedule			
#	Execution time	Deadline	Actions
1	<input type="text" value="5"/>	<input type="text" value="10 December 2019"/>	<a href="#">Delete</a>
2	<input type="text" value="7"/>	<input type="text" value="15 December 2019"/>	<a href="#">Delete</a>
3	<input type="text" value="3"/>	<input type="text" value="25 December 2019"/>	<a href="#">Delete</a>

Execution order:  
**1, 2, 3**

You should start execution at:  
**4th December 2019**

Рисунок 4.2 – Детальний опис вирішення задачі

Користувач має можливість отримати розклад виконання робіт, ввівши дані роботи – директивний термін та тривалість виконання. Екранна форма отримання розкладу зображена на рисунку 4.3.

Create new schedule			
#	Execution time	Deadline	Actions
1	<input type="text" value="3"/>	<input type="text" value="12 November 2019"/>	<button>Delete</button>
2	<input type="text" value="5"/>	<input type="text" value="15 November 2019"/>	<button>Delete</button>
3	<input type="text" value="2"/>	<input type="text" value="15 November 2019"/>	<button>Delete</button>
4	<input type="text" value="8"/>	<input type="text" value="26 November 2019"/>	<button>Delete</button>
5	<input type="text" value="12"/>	<input type="text" value="15 December 2019"/>	<button>Delete</button>
<button>Add work</button>			

Рисунок 4.3 – Отримання нового розкладу

### Висновок до розділу

У даному розділі були наведені засоби розробки програмного забезпечення. Серверна частина була розроблена з використанням мови програмування JavaScript та NodeJS, оскільки дані засоби широко використовуються у розробці серверних частин програмного забезпечення. У якості бази даних була обрана MongoDB із-за швидкості роботи та стабільності. Клієнтська частина була розроблена з використанням бібліотеки React. Програмне забезпечення було написано у середовищі WebStorm. Була описана архітектура програмного забезпечення, що містить базу даних, серверну та клієнтську частину та само забезпечення виконується у веб-переглядачі користувача. Була наведена схема структурна пакетів, що містить 5 пакетів. Програмне забезпечення має 12 компонентів, взаємозв'язок яких був відображений на відповідній схемі структурній компонентів. Були наведені екранні форми програмного забезпечення.

## 5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 5.1 Опис ідеї проекту

Мета: мінімізація сумарного випередження завдань у компанії з максимально пізнім моментом запуску.

Результат: веб застосування, яке дозволить спланувати роботу над проектами таким чином, що початок виконання робіт розпочнеться якомога пізніше.

Опис ідеї стартап-проекту наведений у таблиці 5.1.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Розробити застосування, що дозволить спланувати виконання робіт з метою мінімізації сумарного випередження та максимізації моменту початку виконання робіт.	Використання компанією, що працює за методологією «Канбан», які розробляють програмні продукти.	Виконання завдань точно в строки (“Just In Time”) та мінімізація сумарного очікування між виконання завдань.
	Використання застосування у роботі з декількома станками, що виготовляють певні вироби послідовно та паралельно. Але це не є основним предметним середовищем.	Мінімізація «простою» кожного станку.

Зараз існує багато аналогів, що дозволяють вирішити проблему складання розкладів у сфері планування проектів, але вони не повній мірі відповідають вимогам даної задачі. Наприклад, є методологія ведення проектів «Канбан», яка не в повній мірі вирішує дану проблему.

Вже існує алгоритм, що має назву A1, що дозволяє досить ефективно спланувати робочий процес, але і він не завжди дає найбільш оптимальне рішення і припускає використання тільки одного пристрою.

Визначення сильних, слабких та нейтральних характеристик розробленого алгоритму, порівняно з іншими рішеннями, наведено у таблиці 5.2.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W слабка сторона	N нейтраль на сторона	S сильна сторона
		Проектний варіант	Алгоритм A1	Канбан			
1	Робота з декількома пристроями	1	0	1		+	
2	Період перебування на ринку	0	0	1	+		
3	Коректність спланованого розкладу	1	0.95	0.9			+
4	Середовище візуалізації	1	0	1			+

Таким чином, після аналізу сильних, слабких та нейтральних сторін, що проект має слабку сторону: період перебування на ринку, оскільки проект новий і тільки виходить на ринок. Із сильних сторін можна визначити коректність спланованого розкладу та середовище візуалізації. Нейтральними сторонами є робота з декількома пристроями, що є вимогою задачі.

## 5.2 Технологічний аудит ідеї проекту

Технологічна здійсненність ідей проекту наведена у таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідей проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Застосування, що дозволить спланувати	PHP фреймворк Symfony.	На даний момент часу технологія існує та має останню	Symfony – фреймворк з відкритим вихідним

Продовження таблиці 5.3

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	виконання робіт з метою мінімізації сумарного випередження та максимізації		(стабільну) версію 3.4.15. Також, фреймворк може використовуватись для реалізації математичних алгоритмів.	кодом, яким може користуватися будь-який розробник ПЗ.
2	моменту початку виконання робіт.	Платформа Node.js	На даний момент часу технологія існує та має останню (стабільну) версію 10.14.2. Також, платформа має багато бібліотек, що допоможуть реалізувати алгоритм.	Node.js – це платформа з відкритим вихідним кодом, яку може використовувати будь-який розробник ПЗ.
3		Python фреймворк Flask	На даний момент часу технологія існує та має версію 1.0.2. Python має дуже багато бібліотек для розробки математичних алгоритмів.	Flask – мікрофреймворк, написаний на Python, знаходиться у вільному доступі
Обрана технологія реалізації ідеї проекту: Node.js				

Таким чином, результати технічного аудиту ідеї проекту показали, що проект може бути реалізований за допомогою платформи Node.js, так як вона має велику кількість бібліотек, що дозволяють зручніше працювати з математичними алгоритмами.

### 5.3 Аналіз ринкових можливостей запуску стартап-проекту

Характеристика потенційного ринку стартап-проекту наведена у таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних продуктів, од	4
2	Загальний обсяг продажів, грн	400 в місяць
3	Динаміка ринку (якісна оцінка)	зростає
4	Наявність обмежень для входу (вказати характер обмежень)	немає
5	Специфічні вимоги до стандартизації та сертифікації	необхідна сертифікація як ПЗ для автоматизації управління
6	Середня норма рентабельності в галузі (або по ринку), %	75

Отже, результат аналізу ринкових можливостей запуску проекту показує, що виходження на ринок є досить реальним, але тільки за попереднім оцінювання. Маємо 4 головних продуктів-конкурентів, отже можемо зайняти 20% ринку.

Характеристика потенційних клієнтів стартап-проекту наведена у таблиці 5.5.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Планування розкладу виконання робіт у компанії розробників задля мінімізації «простою»	Компанії, що розробляють програмне забезпечення	Кожна група клієнтів хоче розпочати виконання завдань якомога пізніше та встигнути виконати всі завдання точно в строк	Вимоги до компанії: оперативна підтримка користувачів Вимоги до продукції: коректність планування виконання завдань

Таким чином, проаналізувавши наявних потенційних клієнтів проекту, можна визначити цільову аудиторію, що буде користуватись продуктом – компанії, що розробляють програмне забезпечення. Стартап-проект зі своєї сторони має забезпечити клієнтів оперативною підтримкою та коректність планування.

Фактори загроз стартап-проекту наведені у таблиці 5.6.

Таблиця 5.6 – Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Некоректна робота ПЗ	Як і в будь-якій системі, можуть зустрічатись збої роботи програми, що може зумовити користувача перестати користуватися системою	Оперативне виправлення помилок та підтримка клієнтів
2	Велика конкуренція	Система тільки з'являється на ринку і тому буде складно конкурувати з існуючими рішеннями	Купівля реклами, в якій будуть описані переваги системи над вже існуючими

Таким чином, як і будь-яке програмне забезпечення, може мати некоректну роботу при різних умовах користування. Оперативне вирішення цих помилок допоможе клієнту і далі користуватись системою.

Оскільки система нова і тільки заходить на ринок, то можна очікувати велику конкуренцію з боку вже існуючих рішень. Реклама, що містить прозорий опис функціоналу, може допомогти вирішити цю проблему.

Фактори можливості стартап-проекту наведені у таблиці 5.7.

Таблиця 5.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Швидке захоплення ринку	Система має багато переваг і є перспективною, то може швидко зайняти передові позиції на ринку	І далі покращувати систему та надавати оперативну підтримку клієнтам



Продовження таблиці 5.7

№	Фактор	Зміст можливості	Можлива реакція компанії
2	Створення супровідного ПЗ до вже існуючого	Можна створити ще одну систему, але, наприклад, для використання в іншій сфері	Продовжувати і далі розвивати компанію, та створювати нові продукти

Отже, головними факторами можливості є створення супровідного програмного забезпечення, яке разом із набором функціоналу проекту допомогло б швидко захопити ринок.

Ступеневий аналіз конкуренції на ринку стартап проекту наведений у таблиці 5.8.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентно-спроможною)
За типом конкуренції: чиста	Маємо велику кількість клієнтів, отже відбувається чиста конкуренція	Позитивно, оскільки вплив кожної компанії на ринок є незначним
За рівнем конкурентної боротьби: національна	Спочатку система виходить на ринок однієї країни та конкурує в межах країни	Позитивно, бо дає змогу розвинути на території всієї країни
За галузевою ознакою: внутрішньогалузева	Система конкурує у сфері ведення проєктів	Негативно, оскільки конкурувати та розвиватись система може в межах однієї сфери
Конкуренція за видами товарів: товарно-родова	Системи є схожими, але мають різний функціонал	Позитивно, оскільки система має свій набір функціоналу

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентно-спроможною)
За характером конкурентних переваг: нецінова	Боротьба ведеться за якість наданого функціоналу	Позитивно, оскільки переваги функціоналу на стороні нашої системи. Потрібно продовжувати рекламувати систему.
За інтенсивністю: марочна	Конкуренція на рівні бренду	Позитивно, оскільки маємо можливість розвивати свій бренд та бути впізнаним на ринку

Отже, після виконання ступеневого аналізу конкуренції на ринку, маємо чисту конкуренцію, що зумовлена наявністю великою кількістю клієнтів, має національну конкурентну боротьбу (в межах однієї країни), є внутрішньогалузевою, конкуренція товарно-родова (конкуренція зі схожими системами, які мають різний функціонал), нецінова (конкурентна боротьба ведеться за якість) та за інтенсивністю марочна (конкуренція на рівні бренду).

Аналіз конкуренції в галузі за М. Портером наведений у таблиці 5.9.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Алгоритм A1 та «Канбан»	Щоб увійти в ринок необхідно мати розширений функціонал, ніж у систем, що вже є на ринку	-	Клієнт може обрати товар-субститут, якщо не задоволений якістю системи	Товари-замінники коштують дешевше і тому можуть захопити частину ринку

Продовження таблиці 5.9

Висновки	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Боротьба не буде інтенсивною, оскільки сфера управління проектами є розповсюдженою	Маємо більше функціоналу, отже, є можливість увійти на ринок	—	Клієнти не будуть купувати послуги у незнайомій компанії по завищеним цінам, тому вони диктують правила на ринку	Із-за товарів-замінників є обмеження для роботи на ринку

Обґрунтування факторів конкурентоспроможності наведено у таблиці 5.10.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Оскільки на ринку існує багато схожих систем, то ціна є важливим фактором ведення конкурентної боротьби. Наша ціна складає \$4 в місяць, що є нижчою за ринкову.
2	Репутація компанії	Так як зараз існує велика кількість систем, що дозволяють планувати робочий процес, то репутація компанії є важливим фактором, оскільки рівень довіри клієнта зростає.
3	Частка ринку	Чим більшою часткою ринку володіє компанія, тим більше має ринкові можливості.
4	Маркетинговий бюджет	Здатність створювати рекламу продукту залежить від розміру маркетингового бюджету.

Продовження таблиці 5.10

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
5	Набір наданих послуг	Клієнт має можливість сам обрати набір потрібного функціоналу, не переплачуючи за непотрібний.
6	Якість надання послуг	Окрім кількості послуг також важливим фактором є їх якість.

Розрахунок цін програмного забезпечення на ринку, на розробку системи потрібно 8 місяців. Середня заробітна плата Node.js програміста – \$1400. Обладнання, необхідне для розробки системи – комп'ютери та сервери, середня вартість яких приблизно \$3700, оренда приміщення – \$500/міс., та соціальні внески у розмірі 22%, та інші витрати у розмірі приблизно \$600/міс.

Отже, можна підрахувати ціну програмного забезпечення: \$28628.

Тому, маючи на даний час на ринку України 8456 ІТ-компаній (за даними 2018 року), які у своїй роботі використовують планування виконання робіт за методологією «Канбан» або схожими. На даний час маємо 2 конкурентних продукти, якими користуються дані компанії, та ми можемо претендувати на 33% ринку, тобто, приблизно 2790 компаній.

Отже, кожна компанія повинна заплатити за продукт приблизно \$11, щоб ми покрили всі витрати. Ми хочемо покрити всі ці витрати за 6 місяців, тобто ціна вже \$2/міс ~ 56 грн/міс. Зробимо ціну \$4, щоб швидше покрити всі витрати.

Умовні позначення:

- Алгоритм A1: #
- «Канбан»: &

Порівняльний аналіз сильних та слабких сторін «Системи планування розкладів виконання робіт в області управління проектами» наведений у таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін «Системи планування розкладів виконання робіт в області управління проектами»

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашою системою						
			-3	-2	-1	0	1	2	3
1	Ціна	12		#			&		
2	Репутація компанії	5					#		&
3	Частка ринку	2						#	&
4	Маркетинговий бюджет	5			#			&	
5	Набір наданих послуг	12	&	#					
6	Якість надання послуг	18		&	#				

Отже, після аналізу сильних та слабких сторін продукту, можна виділити відносно невелику ціну, набір та якість наданих послуг як сильна сторона продукту у порівнянні з конкурентами, оскільки маємо плату в місяць нижчу за ринкову. Слабкими сторонами є репутація компанії, оскільки продукт тільки виходить на ринок і мало хто про нього знає, а також розмір маркетингового бюджету. З цього випливає мала частка ринку.

SWOT-аналіз стартап проекту наведений у таблиці 5.12.

Таблиця 5.12 – SWOT-аналіз стартап-проекту

Сильні сторони	Слабкі сторони
Технологічна перевага	Слабке забезпечення фінансами
Якість надання послуг	Мала частка ринку
Продаж послуг через мережу Інтернет	Репутація компанії
Можливості	Загрози
Швидке захоплення ринку	Велика конкуренція
Створення супровідного ПЗ до вже існуючого	Некоректна робота системи

Альтернативи ринкового впровадження стартап-проекту наведені у таблиці 5.13.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створення системи знижок для постійних клієнтів	64%	2.5 міс.
2	Створення можливості інтеграції з іншими сервісами	58%	3 міс.
3	Створення відділу для роботи з клієнтами	86%	1.5 міс.
4	Створення системи збору даних щодо якості надання послуг	46%	2 міс.

Отже, проаналізувавши альтернативи ринкового впровадження проекту та за ймовірністю отримання ресурсів та строками реалізації було обрано «Створення відділу для роботи з клієнтами» як найбільш оптимальну альтернативну поведінку при виходу на ринок.

#### 5.4 Розрахунок економічної ефективності інноваційного проекту за методикою ЮНІДО

Вихідні дані для визначення доходності проекту наведені у таблиці 5.14.

Таблиця 5.14 – Вихідні дані для визначення доходності проекту

№	Показники	Од. вим.	Значення
1	Витрати на обладнання	\$	3700
2	Термін роботи обладнання після вводу	кварталів	8
3	Гарантований обсяг продажів підписок на квартал	\$	33480
4	Квартальний обсяг замовлень	шт.	8370
5	Валютний депозит	%	17
6	Фактор ризику	%	3.2
7	Інфляція на валютному ринку	%	9.1

Поточні витрати на здійснення проекту наведені у таблиці 5.15.

Таблиця 5.15 – Поточні витрати на здійснення проекту

Витрати	1 кв. (1 р.)	2 кв. (1 р.)	3 кв. (1 р.)	4 кв. (1 р.)	1 кв. (2 р.)	2 кв. (2 р.)	3 кв. (2 р.)	4 кв. (2 р.)	Всього
Зарплата	4200	4200	4200	4200	4200	4200	4200	4200	33600
Соціальні нарахування	924	924	924	924	924	924	924	924	7392
Амортизація	232	232	232	232	232	232	232	232	1856
Інші	3800	3800	3800	3800	3800	3800	3800	4200	30400
Всього	9156	9156	9156	9156	9156	9156	9156	9156	73248

#### 5.4.1 Визначення обсягу грошових потоків, дол.

(чистий дохід – собівартість) – (чистий дохід - собівартість)\*0.18 + амортизація

$$1\text{-й квартал (1 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$2\text{-й квартал (1 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$3\text{-й квартал (1 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$4\text{-й квартал (1 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$1\text{-й квартал (1 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$2\text{-й квартал (2 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$3\text{-й квартал (3 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

$$4\text{-й квартал (4 р.)} = (33480 - 9156) * 0.82 + 232 = 20177.68$$

\$28628 – витрати на розробку програмного забезпечення, що враховується в 0-ий квартал.

#### 5.4.2 Визначення норми дисконтування проекту (d)

Норма дисконтування проекту буде рівна:

$$d = 17\%$$

5.4.3 Визначення чистого дисконтованого доходу (ЧДД) та чистої поточної вартості (ЧПВ)

$$\text{ЧДД} = \sum_{t=0}^n \frac{D_t - K_t}{(1+d)^t},$$

де  $D_t$  – чисті доходи  $t$ -го періоду,

$K_t$  – витрати  $t$ -го періоду.

Розрахунок чистого дисконтованого доходу проекту наведений у таблиці 5.16.

Таблиця 5.16 – Розрахунок чистого дисконтованого доходу проекту

Квартали	$D$	$K$	$\frac{1}{(1+d)^t}$	$\frac{D}{(1+d)^t}$	$\frac{K}{(1+d)^t}$	ЧДД	ЧПВ
0	0	28628	1	0	28628	-28628	-28628
1 (1 р.)	20177.68	0	0.854701	17245.88	0	17245.88	-11382.1
2 (1 р.)	20177.68	0	0.730514	14740.07	0	14740.07	3357.949
3 (1 р.)	20177.68	0	0.624371	12598.35	0	12598.35	15956.3
4 (1 р.)	20177.68	0	0.53365	10767.82	0	10767.82	26724.12
1 (2 р.)	20177.68	0	0.456111	9203.265	0	9203.265	35927.38
2 (2 р.)	20177.68	0	0.389839	7866.038	0	7866.038	43793.42
3 (2 р.)	20177.68	0	0.333195	6723.11	0	6723.11	50516.53
4 (2 р.)	20177.68	0	0.284782	5746.248	0	5746.248	56262.78
Всього	161421.44	28628	-	84890.78	28628	67644.9	-

#### 5.4.4 Визначення терміну окупності проекту

Термін окупності проекту ( $T_{ок}$ ) визначається на підставі отриманих раніше значень ЧДД та ЧПВ (таблиця 5.16):

$$T_{ок} = p + \frac{\text{ЧПВ}_p}{\text{ЧДД}_{p+1}},$$

де  $p$  – останній квартал, коли  $\text{ЧПВ} < 0$ ,

$\text{ЧПВ}_p$  – значення ЧПВ в  $p$ -му кварталі (без мінусу),



$\text{ЧДД}_{p+1}$  – значення ЧДД в  $(p + 1)$ -му кварталі.

$$T_{\text{ок}} = 1 + \frac{\text{ЧПВ}_p}{\text{ЧДД}_{p+1}} = 1 + \frac{11382.1}{14740.07} = 1.077 \text{ (квартал)}.$$

Отже, термін окупності проекту дорівнює двом кварталам першого року (оскільки одного буде недостатньо, щоб покрити витрати).

#### 5.4.5 Індекс доходності та середньорічна рентабельність проекту

Індекс доходності (ІД) – це відношення сумарного дисконтованого доходу до сумарних дисконтованих витрат.

$$\text{ІД} = \frac{\sum_{t=0}^n \frac{D}{(1+d)^t}}{\sum_{t=0}^n \frac{K}{(1+d)^t}} = \frac{84890.78}{28628} = 2.965$$

Повинна виконуватись умова  $\text{ІД} > 1$ . Маємо  $2.965 > 1$ , тому проект можна рекомендувати до впровадження по цьому показнику.

Тоді середньорічна рентабельність проекту ( $R$ ) буде:

$$R = \frac{\text{ІД}}{n} * 100\% = \frac{2.965}{8} * 100\% = 37\%$$

Отже, проект є досить рентабельним. Закупівля обладнання відбувається у нульовий квартал, тому реалізація перших замовлень починається вже з першого кварталу. За два квартали компанія вже зможе розрахуватись з інвестором, а наступні шість кварталів буде працювати за власний прибуток. Термін окупності проекту визначається між першим та другим кварталами (коли  $\text{ЧПВ} > 0$ , то проект окупається).

### Висновок до розділу

В ході проведеного аналізу можна зробити висновок, що можливість ринкової комерціалізації проекту є досить високою, оскільки, на даний момент часу існує багато компаній, які у своїй роботі використовують методології ведення проектів та мають бажання покращити планування робочого процесу.

Постійно з'являються нові методології, деякі більш ефективні, деякі менш, тому вихід на ринок даного продукту з функціоналом, що відрізняється від систем, що вже є на ринку, знайде своїх користувачів.

Однак, продукт має і слабкі сторони. Оскільки продукт тільки виходить на ринок, не має достатнього фінансів, бо має низьку репутацію, оскільки, майже ніхто не знає про даний продукт. З цього і витікає мала частка ринку.

Створення відділу для роботи з клієнтами та запровадження системи знижок постійним клієнтам може допомогти швидше зайняти свою нішу на ринку при оперативній підтримці користувачів зі сторони продукту.

Подальша імплементація є важливою, оскільки поява нових технологій може забрати частку ринку даного продукту.

Отже, вихід на ринок даного продукту має велику ймовірність успішної комерційної діяльності, бо існує велика кількість компаній, які б хотіли покращити організацію свого робочого процесу.

## ЗАГАЛЬНІ ВИСНОВКИ

У даній роботі була наведена задача календарного планування з мінімізацією сумарного випередження робіт по директивних термінах та максимізацією моменту початку виконання при виконанні робіт паралельними пристроями.

Були описані типи і методи організації виробництва. Була здійснена порівняльна характеристика різних типів виробництва за різними ознаками: одиничного, серійного та масового. Були наведені методи управління виробництвом: економічні, адміністративно-правові та соціально-психологічні. Оскільки маємо справу з плануванням розкладу робіт, був здійснений опис календарного планування та, зокрема, оперативного-календарного планування, були наведені його основні завдання. Були досліджені наявні системи управління виробництвом, серед яких найбільш популярними в рамках задачі та їй подібній є MRP, MRPII, MES та ERP системи, були описані їх переваги та відмінності. Також, була описана методологія «точно в термін», оскільки в даній задачі необхідно максимізувати момент початку виконання робіт.

Була розглянута проблема планування процесу виконання робіт на підприємстві, був описаний її стан сьогодення. Були наведені технології та методи автоматизації процесів роботи підприємства, а саме календарного планування, зокрема оперативного. Були наведені наявні зараз системи управління виробництвом та здійснене коротке порівняння.

Була наведена постановка задачі та її математична інтерпретація. Були досліджені властивості задачі, були введені необхідні позначення для роботи алгоритму. Були наведені характеристики робіт для подальших експериментальних досліджень. Був наведений приклад розв'язання задачі на конкретних визначених значеннях. Був наведений алгоритм її розв'язку. Роботи були впорядковані таким чином, що змогли досягти мінімізації сумарного випередження директивних термінів робіт та момент початку виконання є максимально можливим значенням.

Задля перевірки ефективності алгоритму було проведено експериментальне дослідження, що підтвердило ефективність алгоритму розв'язання задачі порівняно з методом повного перебору. Задля перевірки ефективності алгоритму задачі були розбиті на підкласи, кожен з яких визначає значення тривалості та директивного терміну роботи. Отримали 27 підкласів задач з різними вхідними параметрами, якими є значення середньої тривалості виконання, дисперсії тривалості та дисперсії директивних строків робіт. Задля кращого розуміння було наведено кодування підкласів задач. Була визначена тенденція погіршення ефективності алгоритму із збільшенням значень дисперсій, але значення ефективності знаходиться у допустимих межах норми в рамках даної задачі.

Програмне забезпечення складається із серверної та клієнтської частин. Серверна частина була розроблена з використанням мови програмування JavaScript та NodeJS, оскільки дані засоби широко використовуються у розробці серверних частин програмного забезпечення. У якості бази даних була обрана MongoDB із-за швидкості роботи та стабільності. Клієнтська частина була розроблена з використанням бібліотеки React. Програмне забезпечення було написано у середовищі WebStorm. Була описана архітектура програмного забезпечення, що містить базу даних, серверну та клієнтську частину та само забезпечення виконується у веб-переглядачі користувача. Була наведена схема структурна пакетів, що містить 5 пакетів. Програмне забезпечення має 12 компонентів, взаємозв'язок яких був відображений на відповідній схемі структурній компонентів. Були наведені екранні форми програмного забезпечення.

В ході роботи був розроблений стартап-проект, в ході аналізу якого був зроблений висновок, що можливість ринкової комерціалізації проекту є досить високою, оскільки, на даний момент часу існує багато компаній, які у своїй роботі використовують методології ведення проектів та мають бажання покращити планування робочого процесу.

Постійно з'являються нові методології, деякі більш ефективні, деякі менш, тому вихід на ринок даного продукту з функціоналом, що відрізняється від систем, що вже є на ринку, знайде своїх користувачів.

Однак, продукт має і слабкі сторони. Оскільки продукт тільки виходить на ринок, не має достатнього фінансів, бо має низьку репутацію, оскільки, майже ніхто не знає про даний продукт. З цього і витікає мала частка ринку.

Створення відділу для роботи з клієнтами та запровадження системи знижок постійним клієнтам може допомогти швидше зайняти свою нішу на ринку при оперативній підтримці користувачів зі сторони продукту.

Подальша імплементація є важливою, оскільки поява нових технологій може забрати частку ринку даного продукту.

Отже, вихід на ринок даного продукту має велику ймовірність успішної комерційної діяльності, бо існує велика кількість компаній, які б хотіли покращити організацію свого робочого процесу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організація виробництва та забезпечення якості продукції [Електронний ресурс] – Режим доступу: [https://pidruchniki.com/10981205/ekonomika/organizatsiya\\_virobnitstva\\_zabezpechennya\\_yakosti\\_produktsiyi](https://pidruchniki.com/10981205/ekonomika/organizatsiya_virobnitstva_zabezpechennya_yakosti_produktsiyi)
2. Типи та методи організації виробництва [Електронний ресурс] – Режим доступу: [https://pidruchniki.com/82234/ekonomika/tipi\\_metodi\\_organizatsiyi\\_virobnitstva](https://pidruchniki.com/82234/ekonomika/tipi_metodi_organizatsiyi_virobnitstva)
3. Методи управління виробництвом [Електронний ресурс] – Режим доступу: [https://studopedia.com.ua/1\\_34224\\_metodi-upravlinnya-virobnitstvom.html](https://studopedia.com.ua/1_34224_metodi-upravlinnya-virobnitstvom.html)
4. Оперативно-календарне планування та контроль на підприємстві [Електронний ресурс] – Режим доступу: [https://pidruchniki.com/86735/finansi/operativno-kalendarne\\_planuvannya\\_kontrol\\_pidpriyemstvi](https://pidruchniki.com/86735/finansi/operativno-kalendarne_planuvannya_kontrol_pidpriyemstvi)
5. Оперативне планування [Електронний ресурс] – Режим доступу: [https://en.wikipedia.org/wiki/Operational\\_planning](https://en.wikipedia.org/wiki/Operational_planning)
6. Автоматизовані системи управління виробництвом: MRP [Електронний ресурс] – Режим доступу: <https://studfiles.net/preview/5734322/page:23/>
7. Автоматизовані системи управління виробництвом: MRP II [Електронний ресурс] – Режим доступу: <https://studfiles.net/preview/5734322/page:24/>
8. MRP II – планування виробничих ресурсів [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/manufacturing-resource-planning-mprii>
9. MES – система управління виробництвом [Електронний ресурс] – Режим доступу: <https://www.it.ua/knowledge-base/technology-innovation/manufacturing-execution-system-mes>
10. Автоматизовані системи управління корпорацією : ERP [Електронний ресурс] – Режим доступу: <https://studfiles.net/preview/5734322/page:25/>

11. Advanced Planning & Scheduling [Электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/Advanced\\_Planning\\_%26\\_Scheduling](https://ru.wikipedia.org/wiki/Advanced_Planning_%26_Scheduling)
12. Методологія Just-in-Time [Электронный ресурс] – Режим доступа: <http://www.up-pro.ru/encyclopedia/just-in-time.html>
13. Why Toyota's Just-in-Time Method is Critical to Its Success [Электронный ресурс] – Режим доступа: <https://articles2.marketrealist.com/2016/05/toyotas-just-time-method-critical-success/>
14. Теорія розкладів – дослідження операцій [Електронний ресурс] – Режим доступу: [http://do.unicyb.kiev.ua/index.php/uk/2011-01-03-16-37-54?task=download&cid\[0\]=34](http://do.unicyb.kiev.ua/index.php/uk/2011-01-03-16-37-54?task=download&cid[0]=34)
15. Классификация задач теории расписаний [Электронный ресурс] – Режим доступа: <https://studfiles.net/preview/6050916/page:2/>
16. Michael L. Pinedo, Scheduling: Theory, Algorithms, and Systems, Springer Publishing Company, Incorporated, 2008
17. Ю.А. Зак – Прикладные задачи теории расписаний и маршрутизации перевозок, ст. 69-72
18. Конвей Р.В., Максвелл В.Л., Миллер Л.В.: – Теория расписаний. – М., Физматгиз, Наука, 1975, 359 с.
19. Танаев В.С., Ковалев М.Я., Шафранский Я.М.: – Теория расписаний. Групповые технологии. – Минск, Институт технической кибернетики НАН Беларуси, 1998, 289 с.
20. Танаев В.С., Гордон В.С., Шафранский Я.М.: – Теория расписаний. Одностадийные системы. – М.: URSS, 1989. 328 с.
21. Domschke W., Scholl A., Voß S.: – Produktionsplanung. Ablauforganisatorische Aspekte. – Berlin, Heidelberg: Springer Verlag, 2005, 456 S.
22. Carlier J.: – The one-machine sequencing problem. – European Journal of Operational Research, 1982, 11, S. 42-47.

23. Johnson S.M.: – Optimal two- and three stage production schedules with setup times included. – Research Logistics Quarterly, 1954, 1, S. 61-68.
24. Brucker P.: – Scheduling Algorithms. – Springer-Verlag, Berlin, Heidelberg und New York 1998.
25. Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment [Електронний ресурс] – Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/0377221794903336>
26. Ant Colony System for JSP [Електронний ресурс] – Режим доступу: [https://link.springer.com/chapter/10.1007/978-3-540-30479-1\\_31](https://link.springer.com/chapter/10.1007/978-3-540-30479-1_31)
27. А.С. Аничкин, В.А. Семенов. Современные модели и методы теории расписаний. . Труды Института системного программирования РАН, том 26, вып. 3, 2014, стр. 5-50. DOI: 10.15514/ISPRAS-2014-26(3)-1.
28. Модели и алгоритмы теории расписаний в задачах планирования и управления проектами / А. А. Павлов, С. К. Чернов, Е. Б. Мисюра // Пр. Одес. політехн. ун-ту. - 2006. - Вип. 1. - С. 150-159. - Библиогр.: 8 назв. - рус.
29. Задача построения допустимого расписания с максимально поздним моментом запуска и минимальным суммарным опережением / М.З. Згуровский, А.А. Павлов, Е.А. Халус // Системні дослідження та інформаційні технології. — 2015. — № 2. — С. 7-15 . — Бібліогр.: 2 назв. — рос.
30. Планирование мелкосерийного производства в АСУП. – М.: Наука, 1973. – С. 16-32.
31. Перовская Е.И. Об одном алгоритме решения задачи календарного планирования // Вычислительные процессы и структуры. – Л.: Машиностроение, 1982. С. 84-92.
32. Евристичні алгоритми розв’язання одноетапних задач складання розкладів для удосконалення моделі багаторівневої системи [Електронний ресурс] – Режим доступу: <https://cyberleninka.ru/article/n/evristichni-algoritmi->



rozvyazannya-odnoetapnih-zadach-skladannya-rozkladiv-dlya-udoskonalennya-modeli-bagatorivnevoyi-sistemi

33. Monma C.L. Analysis of heuristics for preemptive parallel machine scheduling with batch setup times.
34. Lee H. A hybrid bounding procedure for the workload allocation problem on parallel unrelated machines with.
35. Ю.А. Зак – Прикладные задачи теории расписаний и маршрутизации перевозок, ст. 67.
36. Емельянов В. В., Курейчик В. В., Курейчик В. М.: - Теория и практика эволюционного моделирования. – М., Физматлит, Наука, 2003, 434 с.
37. Курейчик В. М.: - Генетические алгоритмы. Монография. – Таганрог: Изд. ТРТУ, 1998, 242 с.
38. Cheveland G.A., Smith S.F.: - Using genetic algorithms to schedule flow shop releases. – In: J.D. Shaffer (Hrsg.): Proceedings of the third international conference on genetic algorithms, Morgan Kaufman, San Mateo, 1989, S. 160-169.
39. Glover, F.: - Tabu Search, Part I, ORSA Journal on Computing 1989 1: 3, p. 190-206; Tabu Search, Part II, ORSA Journal on Computing 1990, 2: 1, p. 4-32.
40. Nissen Volker: - Einführung in Evolutionäre Algorithmen. Optimierung nach dem Vorbild der Evolution, Vieweg, München, 1997, 345 S.
41. Goldberg David E.: - Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
42. Michalewicz Z.: - Heuristic methods for evolutionary computations techniques. – Journal of Heuristics. 1995, 1, p. 177-206.
43. Michalewicz Z.: - Genetic Algorithms + Data Structures = Evolution Programs. – Springer, Berlin, 1999, S. 67.
44. Ващук, Ф.Г., Лавер, О.Г. and Шумило, Н.Я., 2003. Теорія розкладів
45. Эвристические методы календарного планирования. – К.: Техника, 1980. – С. 20-25.

46. Задача построения допустимого расписания с максимально поздним моментом запуска и минимальным суммарным опережением / М. З. Згуровский, А. А. Павлов, Е. А. Халус // Системні дослідження та інформаційні технології. - 2015. - № 2. - С. 7-15. - Режим доступу: [http://nbuv.gov.ua/UJRN/sdtit\\_2015\\_2\\_3](http://nbuv.gov.ua/UJRN/sdtit_2015_2_3)
47. Дослідження задачі визначення максимально пізнього моменту початку виконання робіт з мінімізацією сумарного випередження відносно директивних термінів виконання робіт / Волошин Д.О., Клименко В.М., Жданова О.Г., Сперкач М.О., Халус О.А Київ: МОДС 2019 Тези доповідей, 398-401 ст.
48. Равномерное распределение вероятностей [Електронний ресурс] – Режим доступу: [http://mathprofi.ru/ravnomernoe\\_raspredelenie\\_veroyatnostei.html](http://mathprofi.ru/ravnomernoe_raspredelenie_veroyatnostei.html)
49. Равномерный и нормальный законы распределения непрерывных и случайных величин [Електронний ресурс] – Режим доступу: [http://edu.tltsu.ru/er/book\\_view.php?book\\_id=1cee&page\\_id=19506](http://edu.tltsu.ru/er/book_view.php?book_id=1cee&page_id=19506)
50. Задача складання календарного плану виконання робіт на підприємстві з мінімізацією сумарного випередження директивних термінів та максимізацією моменту початку виконання робіт паралельними пристроями / Волошин Д.О., Жданова О.Г., Сперкач М.О.: Журнал «Інформатика та математичні методи в моделюванні» (подано до друку)
51. Волошин Д.О. Дослідження задачі визначення максимально пізнього моменту початку виконання робіт з мінімізацією сумарного випередження відносно директивних термінів виконання робіт при виконанні робіт паралельними пристроями / М.О. Сперкач, О.Г. Жданова // Матеріали ІІІ всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р. – С. 71-74.
52. JavaScript [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>

53. ECMAScript [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/ECMAScript>
54. React [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/React>
55. AngularJS [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/AngularJS>
56. VueJS [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Vue.js>
57. NodeJS [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Node.js>
58. ElectronJS [Электронный ресурс] – Режим доступа: <https://electronjs.org/>
59. NW.js [Электронный ресурс] – Режим доступа: <https://nwjs.io/>
60. React Native [Электронный ресурс] – Режим доступа:  
<https://facebook.github.io/react-native/>
61. Cordova [Электронный ресурс] – Режим доступа:  
<https://cordova.apache.org/>
62. Adobe Create Suite [Электронный ресурс] – Режим доступа:  
[https://uk.wikipedia.org/wiki/Adobe\\_Creative\\_Suite](https://uk.wikipedia.org/wiki/Adobe_Creative_Suite)
63. JMeter [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/JMeter>
64. Java [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Java>
65. C [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_(programming_language))
66. Self [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Self\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Self_(programming_language))
67. Scheme [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Scheme>

68. CommonJS [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/CommonJS>
69. V8 – JavaScript Engine [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/V8\\_\(JavaScript\\_engine\)](https://en.wikipedia.org/wiki/V8_(JavaScript_engine))
70. Facebook [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Facebook>
71. Instagram [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Instagram>
72. Model-View-Controller [Электронный ресурс] – Режим доступа:  
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
73. DOM [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Document\\_Object\\_Model](https://en.wikipedia.org/wiki/Document_Object_Model)
74. JSX in depth [Электронный ресурс] – Режим доступа:  
<https://reactjs.org/docs/jsx-in-depth.html>
75. PHP [Электронный ресурс] – Режим доступа:  
<https://en.wikipedia.org/wiki/PHP>
76. MongoDB [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/MongoDB>
77. Document-oriented database [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Document-oriented\\_database](https://en.wikipedia.org/wiki/Document-oriented_database)
78. Relational database [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Relational\\_database#RDBM](https://en.wikipedia.org/wiki/Relational_database#RDBM)
79. JSON [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/JSON>
80. MapReduce [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/MapReduce>
81. BSON [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/BSON>

82. MongoDB – Why schemaless? [Электронный ресурс] – Режим доступа:  
<https://www.mongodb.com/blog/post/why-schemaless>
83. WebStorm [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/WebStorm>
84. HTML [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/HTML>
85. CSS [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/CSS>
86. JetBrains [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/JetBrains>
87. IntelliJ IDEA [Электронный ресурс] – Режим доступа:  
[https://uk.wikipedia.org/wiki/IntelliJ\\_IDEA](https://uk.wikipedia.org/wiki/IntelliJ_IDEA)
88. PhpStorm [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/PhpStorm>
89. CoffeeScript [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/CoffeeScript>
90. TypeScript [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/TypeScript>
91. Code refactoring [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Code\\_refactoring](https://en.wikipedia.org/wiki/Code_refactoring)
92. Dart [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/Dart>
93. Debugging [Электронный ресурс] – Режим доступа:  
<https://en.wikipedia.org/wiki/Debugging>
94. Version control [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control)
95. World Wide Web [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/World\\_Wide\\_Web](https://en.wikipedia.org/wiki/World_Wide_Web)

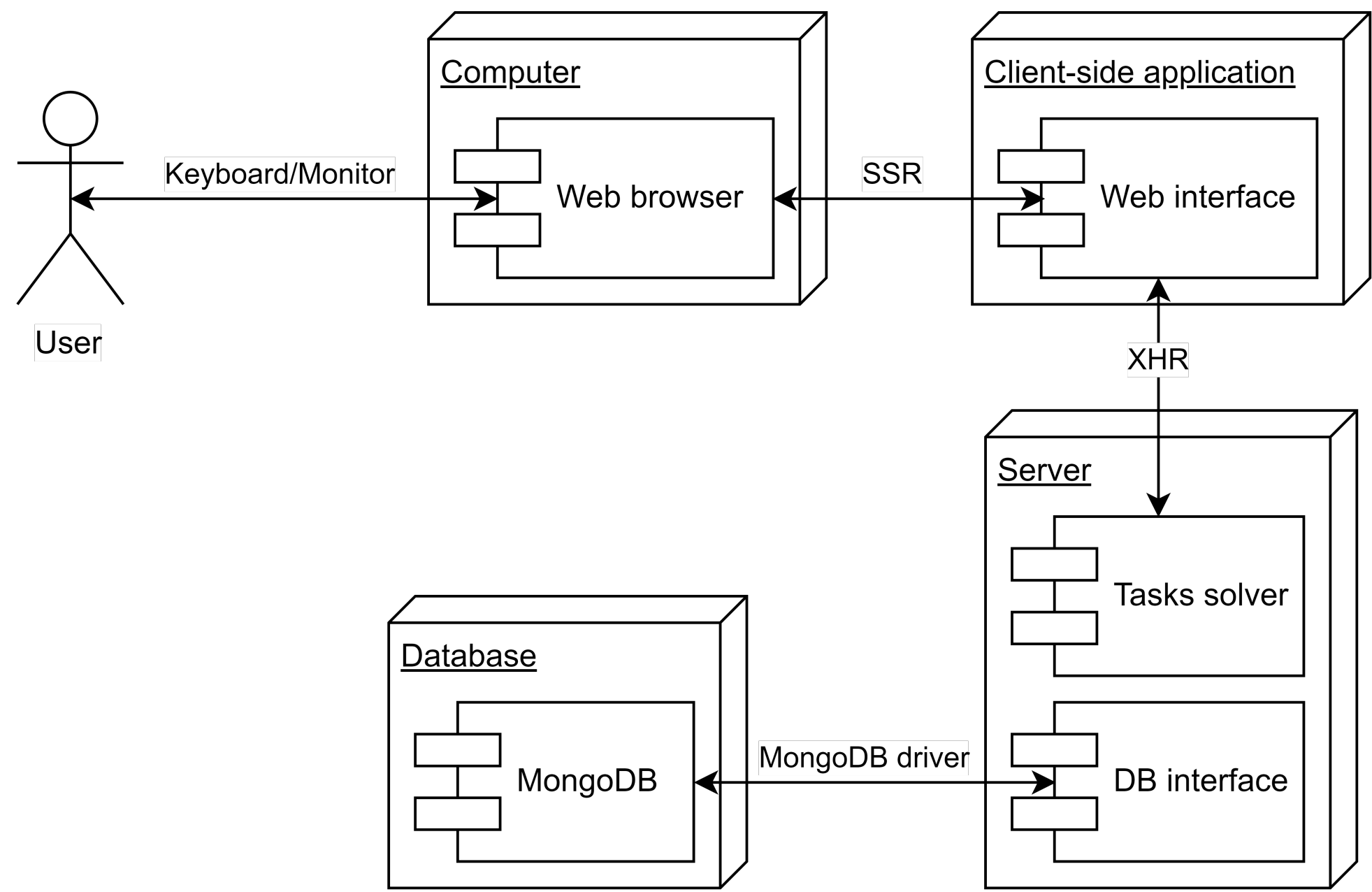
96. W3C [Электронный ресурс] – Режим доступа:  
<https://uk.wikipedia.org/wiki/W3C>
97. Client-Server model [Электронный ресурс] – Режим доступа:  
[https://en.wikipedia.org/wiki/Client%E2%80%93server\\_model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model)

## **ДОДАТКИ**

### **ДОДАТОК А**

#### **Графічний матеріал**

# Плакат 1. Схема структурна розгортання



Демонстраційний плакат до магістерської дисертації  
на тему «Складання календарного плану виконання робіт з мінімізацією сумарного випередження  
директивного строку при виконанні робіт паралельними пристроями»

Виконав студент гр. ІС-81мп

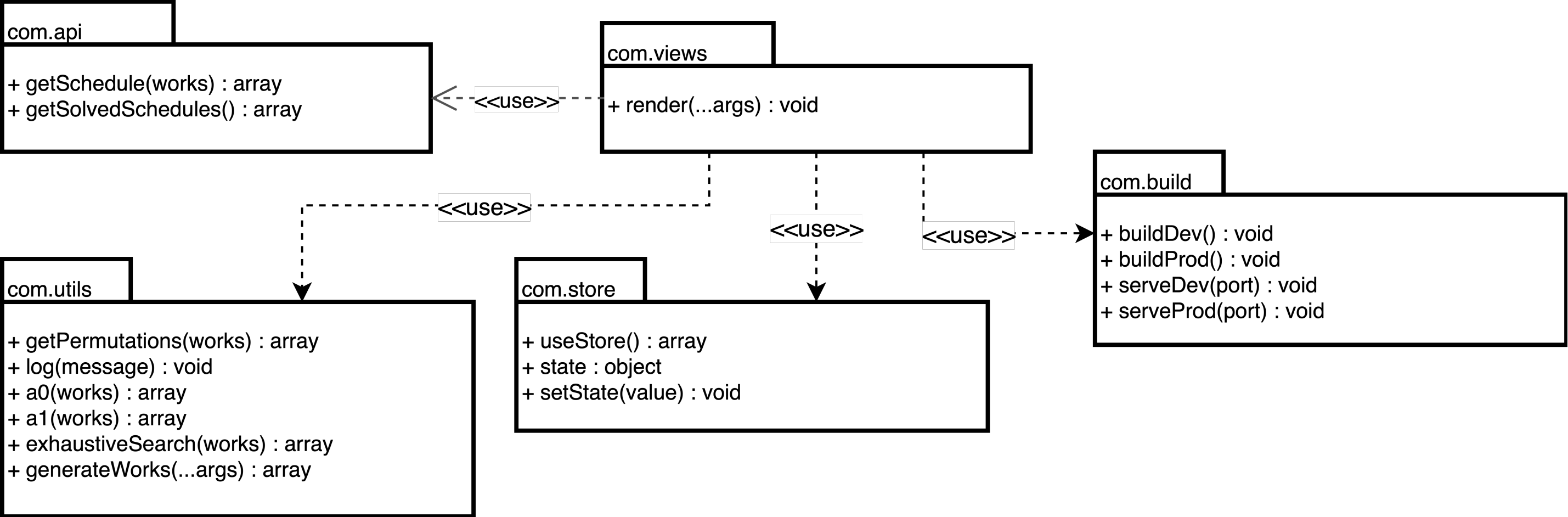
Волошин Д.О.

Керівник

Сперкач М.О.



# Плакат 2. Схема структурна пакетів



Демонстраційний плакат до магістерської дисертації  
на тему «Складання календарного плану виконання робіт з мінімізацією сумарного випередження  
директивного строку при виконанні робіт паралельними пристроями»

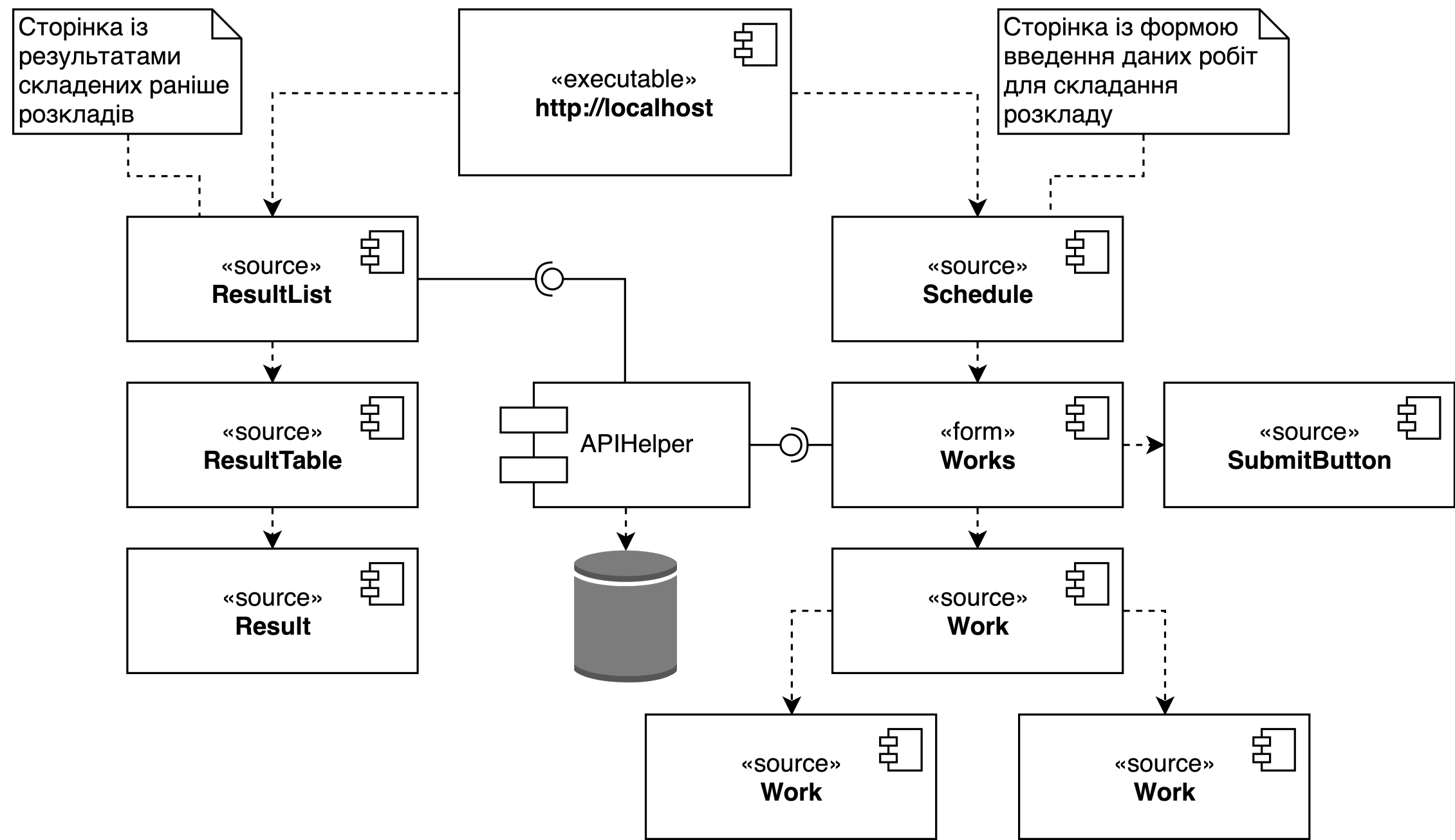
Виконав студент гр. ІС-81мп

Волошин Д.О.

Керівник

Сперкач М.О.

# Плакат 3. Схема структурна компонентів



Демонстраційний плакат до магістерської дисертації  
на тему «Складання календарного плану виконання робіт з мінімізацією сумарного випередження  
директивного строку при виконанні робіт паралельними пристроями»

Виконав студент гр. ІС-81мп

Волошин Д.О.

Керівник

Сперкач М.О.